ENCYCLOPEDIA OF
ARTIFICIAL INTELLIGENCE

Juan Ramón Rabuñal Dopico, Julián Dorado de la Calle, & Alejandro Pazos Sierra

Encyclopedia of Artificial Intelligence

Juan Ramón Rabuñal Dopico
University of A Coruña, Spain

Julián Dorado de la Calle
University of A Coruña, Spain

Alejandro Pazos Sierra
University of A Coruña, Spain

Editorial Advisory Board

List of Contributors

Contents
by Volume

Volume II

Volume III

Preface

Through the history the man has always hoped the boost of three main characteristics: physical, metaphysical and intellectual.

From the physical viewpoint he invented and developed all kind of tools: levers, wheels, cams, pistons, etc., until achieving the sophisticated machines existing nowadays.

Regarding the metaphysical aspect, the initial celebration of magical-animistic rituals led to attempts, either real or literary, for creating ex nihilo life: life from inert substance. The most actual approaches involve the cryoconservation of deceased people for them to be returned to life in the future; the generation of life at the laboratories by means of cells, tissues, organs, systems or individuals created from previously frozen stem cells is also currently aimed.

The third aspect considered, the intellectual one, is the most interesting here. There have been multiple contributions, since devices that increased the calculi ability as the abacus appeared, until the later theoretical proposals for trying to solve problems, as the Ars Magna by Ramón Lull. The first written reference of the Artificial Intelligence that is known is The Iliad, where Homer describes the visit of the goddess Thetis and her son Achilles to the workshop of Hephaestus, god of smiths: At once he was helped along by female servants made of gold, who moved to him. They look like living servant girls, possessing minds, hearts with intelligence, vocal chords, and strength.

However, the first reference of Artificial Intelligence, as it is currently understood, can be found in the proposal made by J. McCarthy to the Rockefeller Foundation in 1956; this proposal hoped for funds that might support a month-lasting meeting of twelve researchers of the Dartmouth Summer Research Project in order to establish the basis of the, McCarthy-named, Artificial Intelligence.

Although the precursors of the Artificial Intelligence (S. Ramón y Cajal, N. Wienner, D. Hebb, C. Shannon and J. McCulloch, among many others), come from multiple science disciplines, the true driving forces (A. Turing, J. von Neumann, M. Minsky, T. Gödell,…) emerge in the second third of the XX century with the apparition of certain tools, the computers, capable of handling fairly complex problems. Some other scientists, as J. Hopfield or J. Holland, proposed at the last third of the century some biology-inspired approaches that enabled the treatment of complex problems of the real world that even might require certain adaptive ability.

All this long and productive trend of the history of the Artificial Intelligence demanded an encyclopaedia that might give expression to the current situation of this multidisciplinary topic, where researches from multiple fields as neuroscience, computing science, cognitive sciences, exact sciences and different engineering areas converge.

This work intends to provide a wide and well balanced coverage of all the points of interest that currently exist in the field of Artificial Intelligence, from the most theoretical fundamentals to the most recent industrial applications.

Multiple researches have been contacted and several notifications have been performed in different forums of the scientific field dealt here.

All the proposals have been carefully revised by the editors for balancing, as far as possible, the contributions, with the intention of achieving an accurately wide document that might exemplify this field.

A first selection was performed after the reception of all the proposals and it was later sent to three external expert reviewers in order to carry out a double-blind revision based on a peer review. As a result of this strict and complex process, and before the final acceptance, a high number of contributions (80% approximately) were rejected or required to be modified.

The effort of the last two years is now believed to be worthwhile; at least this is the belief of the editors who, with the invaluable help of a high number of people mentioned in the acknowledgements, have managed to get this complete encyclopaedia off the ground. The numbers speak for themselves: 233 articles published that have been carried out by 442 authors from 38 different countries and also revised by 238 scientific reviewers. The diverse and comprehensive coverage of the disciplines directly related with the Artificial Intelligence is also believed to contribute to a better understanding of all the researching related to this important field of study. It was also intended that the contributions compiled in this work might have a considerable impact on the expansion and the development of the body of knowledge related to this wide field, for it to be an important reference source used by researchers and system developers of this area. It was hoped that the encyclopaedia might be an effective help in order to achieve a better understanding of concepts, problems, trends, challenges and opportunities related to this field of study; it should be useful for the research colleagues, for the teaching personnel, for the students, etc. The editors will be happy to know that this work could inspire the readers for contributing to new advances and discoveries in this fantastic work area that might themselves also contribute to a better life quality of different society aspects: productive processes, health care or any other area where a system or product developed by techniques and procedures of Artificial Intelligence might be used.

About the Editors

Juan Ramón Rabuñal Dopico is associate professor in the Department of Information and Communications Technologies, University of A Coruña (Spain). He finished his graduate in computer science in 1996, and in 2002, he became a PhD in computer science with his thesis "Methodology for the Development of Knowledge Extraction Systems in ANNs" and he became a PhD in civil engineering in 2008. He has worked on several Spanish and European projects and has published many books and papers in several international journals. He is currently working in the areas of evolutionary computation, artificial neural networks, and knowledge extraction systems.

Julian Dorado is associate professor in the Faculty of Computer Science, University of A Coruña (Spain). He finished his graduate in computer science in 1994. In 1999, he became a PhD, with a special mention of European doctor. In 2004, he finished his graduate in biology. He has worked as a teacher of the university for more than 8 years. He has published many books and papers in several journals and international conferences. He is presently working on bioinformatics, evolutionary computing, artificial neural networks, computer graphics, and data mining.

Alejandro Pazos is professor in computer science, University of A Coruña (Spain). He was born in Padron in 1959. He is MD by Faculty of Medicine, University of Santiago de Compostela in 1987. He obtained a Master of Knowledge Engerineering in 1989 and a PhD in computer science in 1990 from the Polytechnique University of Madrid. He also archives the PhD grade in Medicine in 1996 by the University Complutese of Madrid. He has worked with research groups at Georgia Institute of Technology, Havard Medical School, Stanford University, Politechnique University of Madrid, etc. He funded and is the director of the research laboratory Artificial Neural Networks and Adaptative Systems in Computer science Faculty and is co-director of the Medical Informatics and Radiology Diagnostic Center at the University of A Coruña.

Active Learning with SVM

Jun Jiang
City University of Hong Kong, Hong Kong

Horace H. S. Ip
City University of Hong Kong, Hong Kong

INTRODUCTION

With the increasing demand of multimedia information retrieval, such as image and video retrieval from the Web, there is a need to find ways to train a classifier when the training dataset is combined with a small number of labelled data and a large number of unlabeled one. Traditional supervised or unsupervised learning methods are not suited to solving such problems particularly when the problem is associated with data in a high-dimension space. In recent years, many methods have been proposed that can be broadly divided into two groups: **semi-supervised** and **active learning** (AL). Support Vector Machine (SVM) has been recognized as an efficient tool to deal with high-dimensionality problems, a number of researchers have proposed algorithms of Active Learning with SVM (ALSVM) since the turn of the Century. Considering their rapid development, we review, in this chapter, the state-of-the-art of ALSVM for solving classification problems.

BACKGROUND

The general framework of AL can be described as in Figure 1. It can be seen clearly that its name – **active learning** – comes from the fact that the learner can improve the classifier by actively choosing the "optimal" data from the potential query set Q and adding it into the current labeled training set L after getting its label during the processes. The key point of AL is its sample selection criteria.

AL in the past was mainly used together with neural network algorithm and other learning algorithms. Statistical AL is one classical method, in which the sample minimizing either the variance (D. A. Cohn, Ghahramani, & Jordan, 1996), bias (D. A. Cohn, 1997) or generalisation error (Roy & McCallum, 2001) is queried to the oracle. Although these methods have strong theoretical foundation, there are two common problems limiting their application: one is how to estimate the posterior distribution of the samples, and the other is its prohibitively high computation cost. To deal with the above two problems, a series of **version space based AL** methods, which are based on the assumption that the target function can be perfectly expressed by one hypothesis in the version space and in which the sample that can reduce the volume of the version space is chosen, have been proposed. Examples are query by committee (Freund, Seung, Shamir, & Tishby, 1997), and SG AL (D. Cohn, Atlas, & Ladner, 1994). However the complexity of version space made them intractable until the version space based ALSVMs have emerged.

The success of SVM in the 90s has prompted researchers to combine AL with SVM to deal with the semi-supervised learning problems, such as distance-based (Tong & Koller, 2001), RETIN (Gosselin & Cord, 2004) and Multi-view (Cheng & Wang, 2007) based ALSVMs. In the following sections, we summarize existing well-known ALSVMs under the framework of **version space theory**, and then briefly describe some mixed strategies. Lastly, we will discuss the research trends for ALSVM and give conclusions for the chapter.

VERSION SPACE BASED ACTIVE LEARNING WITH SVM

The idea of almost all existing heuristic ALSVMs is explicitly or implicitly to find the sample which can reduce the volume of the **version space**. In this section, we first introduce their theoretical foundation and then review some typical ALSVMs.

Figure 1. Framework of active learning

Initialize Step: An classifier *h* is trained on the initial labeled training set *L*

step 1: The learner evaluates each data *x* in potential query set *Q* (subset of or whole unlabeled data set *U*) and query the sample *x** which has lowest *EvalFun(x, L, h, H)* to the oracle and get its label *y**;

step 2: The learner update the classifier *h* with the enlarged training set {*L* + (*x**, *y**)};

step 3: Repeat step 1 and 2 until stopping training;

Where

➢   *EvalFun(x, L, h, H)*: the function of evaluating potential query *x* (the lowest value is the best here)

➢   *L*: the current labeled training set

➢   *H*: the hypothesis space

Version Space Theory

Based on the Probability Approximation Correct learning model, the goal of machine learning is to find a consistent classifier which has the lowest generalization error bound. The Gibbs generalization error bound (McAllester, 1998) is defined as

$$\varepsilon_{Gibbs}\left(m, P_H, z, \delta \right) = \frac{1}{m}\left(\ln\left(\frac{1}{P_H\left(V(z)\right)} \right) \right) + \ln\left(\frac{em^2}{\delta} \right)$$

where P_H denotes a prior distribution over hypothesis space *H*, *V(z)* denotes the version space of the training set *z*, *m* is the number of *z* and δ is a constant in [0, 1]. It follows that the generalization error bound of the consistent classifiers is controlled by the volume of the version space if the distribution of the version space is uniform. This provides a theoretical justification for version space based ALSVMs.

Query by Committee with SVM

This algorithm was proposed by (Freund et al., 1997) in which 2*k* classifiers were randomly sampled and the sample on which these classifiers have maximal disagreement can approximately halve the **version space** and then will be queried to the oracle. However, the complexity of the structure of the version space leads to the difficulty of random sampling within it.

(Warmuth, Ratsch, Mathieson, Liao, & Lemmem, 2003) successfully applied the algorithm of playing billiard to randomly sample the classifiers in the SVM version space and the experiments showed that its performance was comparable to the performance of **standard distance-based ALSVM** (SD-ALSVM) which will be introduced later. The deficiency is that the processes are time-consuming.

Standard Distance Based Active Learning with SVM

For SVM, the **version space** can be defined as:

$$V = \left\{ w \in W \mid \|w\| = 1, \quad y_i(w \bullet \Phi(x_i) > 0, \quad i = 1,...,m \right\}$$

where $\Phi(.)$ denotes the function which map the original input space *X* into a high-dimensional space $\Phi(X)$, and *W* denotes the parameter space. SVM has two properties which lead to its tractability with AL. The first is its duality property that each point *w* in *V* corresponds to one hyperplane in $\Phi(X)$ which divides $\Phi(X)$ into two parts and vice versa. The other property is that the solution of SVM w^* is the center of the **version space** when the version space is symmetric or near to its center when it is asymmetric.

Based on the above two properties, (Tong & Koller, 2001) inferred a lemma that the sample nearest to the

Figure 2. Illustration of standard distance-based ALSVM
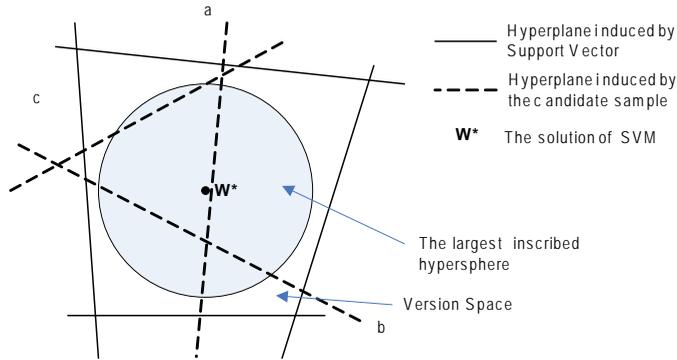
Figure 2a. The projection of the parameter space around the Version Space
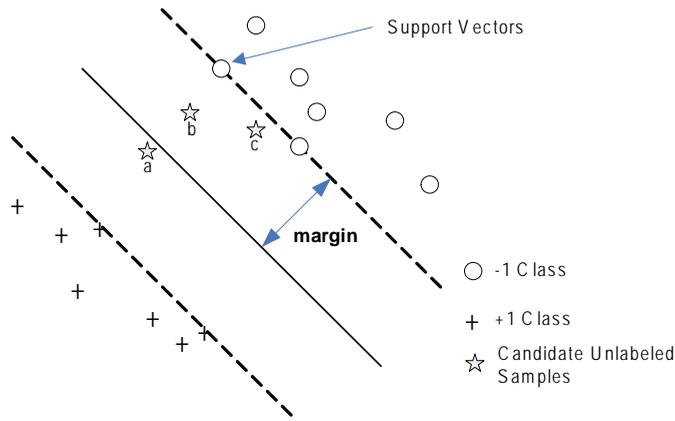
Figure 2b. In the induced feature space

decision boundary can make the expected size of the version space decrease fastest. Thus the sample nearest to the decision boundary will be queried to the oracle (Figure 2). This is the so-called SD-ALSVM which has low additional computations for selecting the queried sample and fine performance in real applications.

Batch Running Mode Distance Based Active Learning with SVM

When utilizing batch query, (Tong & Koller, 2001) simply selected multiple samples which are nearest to the decision boundary. However, adding a batch of such samples cannot ensure the largest reduction of the size of version space, such as an example shown in figure 3. Although every sample can nearly halve the version space, three samples together can still reduce about 1/2,

instead of 7/8, of the size of the version space. It can be observed that this was ascribed to the small angles between their induced hyperplanes.

To overcome this problem, (Brinker, 2003) proposed a new selection strategy by incorporating **diversity** measure that considers the angles between the induced hyperplanes. Let the labeled set be L and the pool query set be Q in the current round, then based on the diversity criterion the further added sample x_q should be

$$x_q = \min_{x_j \in Q} \max_{x_i \in L} \frac{\left|k(x_j, x_i)\right|}{\sqrt{k(x_j, x_j)k(x_i, x_i)}}$$

$$\frac{\left|k(x_j, x_i)\right|}{\sqrt{k(x_j, x_j)k(x_i, x_i)}}$$

3

Figure 3. One example of simple batch querying with "a", "b" and "c" samples with pure SD-ALSVM
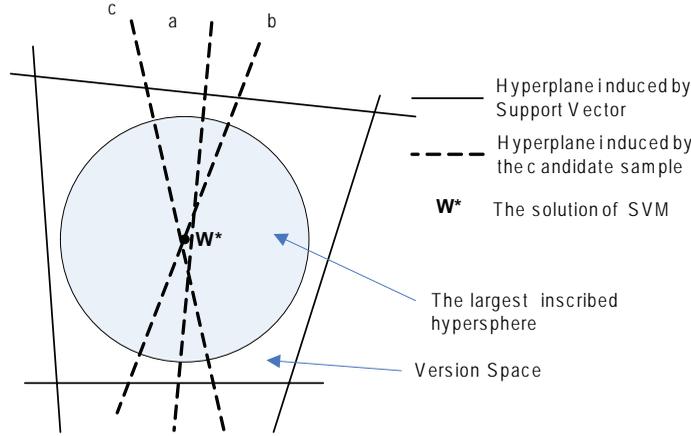
Figure 4. One example of batch querying with "a", "b" and "c" samples by incorporating diversity into SD-ALSVM
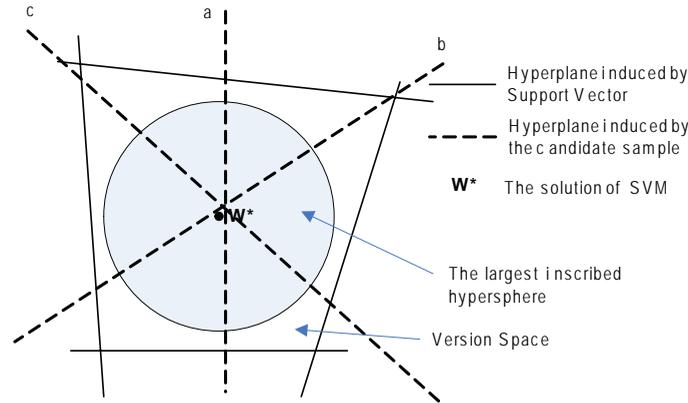
where denotes the cosine value of the angle between two hyperplanes induced by x_j and x_i, thus it is known as angle diversity criterion. It can be observed that the reduced volume of the version space in figure 4 is larger than that in Figure 3.

RETIN Active Learning

Let $(I_j)_{j\in[1...n]}$ be the samples in a potential query set Q, and $r(i, k)$ be the function that, at iteration i, codes the position k in the relevance ranking according to the distance to the current decision boundary, then a sequence can be obtained as follows:

$$\underbrace{I_{r(i,1)}}_{most\ relevant}, I_{r(i,2)},..., \underbrace{I_{r(i,s(i))},..., I_{r(i,s(i)+m-1)}}_{queried\ data},..., \underbrace{I_{r(i,n)}}_{least\ relevant}$$

In SD-ALSVM, $s(i)$ is such as $I_{r(i,s(i))},..., I_{r(i,s(i)+m-1)}$ are the *m* closest samples to the SVM boundary. This strategy implicitly relies on a strong assumption: an accurate estimation of SVM boundary. However, the decision boundary is usually unstable at the initial iterations. (Gosselin & Cord, 2004) noticed that, even if the decision boundary may change a lot during the earlier iterations, the ranking function $r()$ is quite stable. Thus they proposed a balanced selection criterion that

is independent on the frontier and in which an adaptive method was designed to tune *s* during the feedback iterations. It was expressed by

$$s(i+1) = s(i) + h(r_{rel}(i) \; r_{irr}(i)$$

where $h(x, y) = k \times (x - y)$ which characterizes the system dynamics (k is a positive constant), $r_{rel}(i)$ and $r_{irrl}(i)$ denote the number of relevant and irrelevant samples in the queried set in the *ith* iteration. This way, the number of relevant and irrelavant samples in the queried set will be roughly equal.

Mean Version Space Criterion

(He, Li, Zhang, Tong, & Zhang, 2004) proposed a selection criterion by minimizing the **mean version space** which is defined as

$$C_{MVS}(x_k) = Vol(V_i^+(x_k) \; P(y_k = 1 \mid x_k) + Vol(V_i^-(x_k) \; P(y_k = -1 \mid x_k)$$

where $Vol(V_i^+(x_k) \; (Vol(V_i^-(x_k))$ denotes the volume of the version space after adding an unlabelled sample x_k into the *ith* round training set. The mean version space includes both the volume of the version space and the posterior probabilities. Thus they considered that the criterion is better than the SD-ALSVM. However, the computation of this method is time-consuming.

Multi-View Based Active Learning

Different from the algorithms which are based only on one whole feature set, **multi-view** methods are based on multiple sub-feature ones. Several classifiers are first trained on different sub-feature sets. Then the samples on which the classifiers have the largest disagreements comprise the contention set from which queried samples are selected. first (I. Muslea, Minton, & Knoblock, 2000) applied in AL and (Cheng & Wang, 2007) implemented it with ALSVM to produce a Co-SVM algorithm which was reported to have better performance than the SD-ALSVM.

Multiple classifiers can find the rare samples because they observe the samples with different views. Such property is very useful to find the diverse parts belonging to the same category. However, multi-view based methods demand that the relevant classifier can classify the samples well and that all feature sets are uncorrelated. It is difficult to ensure this condition in real applications.

MIXED ACTIVE LEARNING

Instead of single AL strategies in the former sections, we will discuss two mixed AL modes in this section: one is combining different selection criteria and another is incorporating **semi-supervised learning** into AL.

Hybrid Active Learning

Contrast to developing a new AL algorithm that works well for all situations, some researchers argued that combining different methods, which are usually complementary, is a better way, for each method has its advantages and disadvantages. The intuitive structure of the hybrid strategy is parallel mode. The key point here is how to set the weights of different AL methods.

The simplest way is to set fixed weights according to experience and it was used by most existing methods. The Most Relevant/Irrelevant (L. Zhang, Lin, & Zhang, 2001) strategies can help to stabilize the decision boundary, but have low learning rates; while standard distance-based methods have high learning rates, but have unstable frontiers at the initial feedbacks. Considering this, (Xu, Xu, Yu, & Tresp, 2003) combined these two strategies to achieve better performance than only using a single strategy. As stated before, the **diversity** and distance-based strategies are also complementary and (Brinker, 2003), (Ferecatu, Crucianu, & Boujemaa, 2004) and (Dagli, Rajaram, & Huang, 2006) combined angle, inner product and entropy **diversity** strategy with standard distance-based one respectively.

However, the strategy of the fixed weights can not fit well into all datasets and all learning iterations. So the weights should be set dynamically. In (Baram, El-Yaniv, & Luz, 2004), all the weights were initialized with the same value, and were modified in the later iterations by using EXP4 algorithm. In this way, the resulting AL algorithm is empirically shown to consistently perform almost as well as and sometimes outperform the best algorithm in the ensemble.

Semi-Supervised Active Learning

1. Active Learning with Transductive SVM

In the first stages of SD-ALSVM, a few labeled data may lead to great deviation of the current solution from the true solution; while if unlabeled samples are considered, the solution may be closer to the true solution. (Wang, Chan, & Zhang, 2003) showed that the closer the current solution is to the true one, the larger the size of the version space will be reduced. They incorporated Transductive SVM (TSVM) to produce more accurate intermediate solutions. However, several studies (T. Zhang & Oles, 2000) challenged that TSVM might not be so helpful from unlabeled data in theory and in practice. (Hoi & Lyu, 2005) applied the semi-supervised learning techniques based on the Gaussian fields and Harmonic functions instead and the improvements were reported to be significant.

2. Incorporating EM into Active Learning

(McCallum & Nigam, 1998) combined Expectation Maximization (EM) with the strategy of querying by committee. And (Ion Muslea, Minton, & Knoblock, 2002) integrated Multi-view AL algorithm with EM to get the Co-EMT algorithm which can work well in the situation where the views are incompatible and correlated.

FUTURE TRENDS

How to Start the Active Learning

AL can be regarded as the problem of searching target function in the version space, so a good initial classifier is important. When the objective category is diverse, the initial classifier becomes more important, for bad one may result in converging to a local optimal solution, i.e., some parts of the objective category may not be correctly covered by the final classifier. Two-stage (Cord, Gosselin, & Philipp-Foliguet, 2007), long-term learning (Yin, Bhanu, Chang, & Dong, 2005), and pre-cluster (Engelbrecht & BRITS, 2002) strategies are promising.

Feature-Based Active Learning

In AL, the feedback from the oracle can also help to identify the important features, and (Raghavan, Madani, & Jones, 2006) showed that such works can improve the performance of the final classifier significantly. In (Su, Li, & Zhang, 2001), Principal Components Analysis was used to identify important features. To our knowledge, there are few reports addressing the issue.

The Scaling of Active Learning

The scaling of AL to very large database has not been extensively studied yet. However, it is an important issue for many real applications. Some approaches have been proposed on how to index database (Lai, Goh, & Chang, 2004) and how to overcome the concept complexities accompanied with the scalability of the dataset (Panda, Goh, & Chang, 2006).

CONCLUSION

In this chapter, we summarize the techniques of ALSVM which have been an area of active research since 2000. We first focus on the descriptions of heuristic ALSVM approaches within the framework of the theory of version space minimization. Then mixed methods which can complement the deficiencies of single ones are introduced and finally future research trends focus on techniques for selecting the initial labeled training set, feature-based AL and the scaling of AL to very large database.

REFERENCES

Baram, Y., El-Yaniv, R., & Luz, K. (2004). Online Choice of Active Learning Algorithms. *Journal of Machine Learning Research, 5*, 255-291.

Brinker, K. (2003). *Incorporating Diversity in Active Learning with Support Vector Machines.* Paper presented at the International Conference on Machine Learning.

Cheng, J., & Wang, K. (2007). Active learning for image retrieval with Co-SVM. *Pattern Recognition, 40*(1), 330-334.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving Generalization with Active Learning. *Machine Learning, 15*, 201-221.

Cohn, D. A. (1997). Minimizing Statistical Bias with Queries. In *Advances in Neural Information Processing Systems 9, Also appears as AI Lab Memo 1552, CBCL Paper 124*. M. Mozer et al, eds.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active Learning with Statistical Models. *Journal of Artificial Intelligence Research, 4*, 129-145.

Cord, M., Gosselin, P. H., & Philipp-Foliguet, S. (2007). Stochastic exploration and active learning for image retrieval. *Image and Vision Computing, 25*(1), 14-23.

Dagli, C. K., Rajaram, S., & Huang, T. S. (2006). *Utilizing Information Theoretic Theoretic Diversity for SVM Active Learning*. Paper presented at the International Conference on Pattern Recognition, Hong Kong.

Engelbrecht, A. P., & BRITS, R. (2002). Supervised Training Using an Unsuerpvised Approach to Active Learning. *Neural Processing Letters, 15*, 14.

Ferecatu, M., Crucianu, M., & Boujemaa, N. (2004). *Reducing the redundancy in the selection of samples for SVM-based relevance feedback*

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective Sampling Using the Query by Committee Algorithm. *Machine Learning, 28*, 133-168.

Gosselin, P. H., & Cord, M. (2004). *RETIN AL: an active learning strategy for image category retrieval*. Paper presented at the International Conference on Image Processing.

He, J., Li, M., Zhang, H.-J., Tong, H., & Zhang, C. (2004). *Mean version space: a new active learning method for content-based image retrieval*. Paper presented at the International Multimedia Conference Proceedings of the 6th ACM SIGMM International Workshop on Mulitimedia Information Retrieval.

Hoi, S. C. H., & Lyu, M. R. (2005). *A semi-supervised active learning framework for image retrieval*. Paper presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Lai, W.-C., Goh, K., & Chang, E. Y. (2004, June). *On Scalability of Active Learning for Formulating Query Concepts (long version of the ICME invited paper)*.

Paper presented at the Workshop on Computer Vision Meets Databases (CVDB) in cooperation with ACM International Conference on Management of Data (SIGMOD), Paris.

McAllester, D. A. (1998). *Some PAC Bayesian Theorems*. Paper presented at the Proceedings of the 11th Annual Conference on Computational Learning Theory, Madison, Wisconsin.

McCallum, A. K., & Nigam, K. (1998). *Employing EM and Pool-Based Active Learning for Text Classification*. Paper presented at the Proceedings of 15th International Conference on Machine Learning.

Muslea, I., Minton, S., & Knoblock, C. A. (2000). *Selective Sampling with Redundant Views*. Paper presented at the Proceedings of the 17th National Conference on Artificial Intelligence.

Muslea, I., Minton, S., & Knoblock, C. A. (2002). *Active+Semi-Supervised Learning = Robust Multi-View Learning*. Paper presented at the Proceedings of the 19th International Conference on Machine Learning.

Panda, N., Goh, K., & Chang, E. Y. (2006). Active Learning in Very Large Image Databases *Journal of Multimedia Tools and Applications Special Issue on Computer Vision Meets Databases*.

Raghavan, H., Madani, O., & Jones, R. (2006). Active Learning with Feedback on Both Features and Instances. *Journal of Machine Learning Research, 7*, 1655-1686.

Roy, N., & McCallum, A. (2001). *Toward Optimal Active Learning Through Sampling Estimation of Error Reduction*. Paper presented at the Proceedings of 18th International Conference on Machine Learning.

Su, Z., Li, S., & Zhang, H. (2001). *Extraction of Feature Subspaces for Content-based Retrieval Using Relevance Feedback*. Paper presented at the ACM Multimedia, Ottawa, Ontario, Canada.

Tong, S., & Koller, D. (2001). Support Vector Machine Active Learning with Application to Text Classification. *Journal of Machine Learning Research*, 45-66.

Wang, L., Chan, K. L., & Zhang, Z. (2003). *Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval*. Paper presented at the Proceeding of IEEE Computer Vision and Pattern Recognition.

A

Warmuth, M. K., Ratsch, G., Mathieson, M., Liao, J., & Lemmem, C. (2003). Active Learning in the Drug Discovery Process. *Journal of Chemical Information Sciences, 43*(2), 667-673.

Xu, Z., Xu, X., Yu, K., & Tresp, V. (2003). *A Hybrid Relevance-feedback Approach to Text Retrieval.* Paper presented at the Proceedings of the 25th European Conference on Information Retrieval Research, Lecture Notes in Computer Science.

Yin, P., Bhanu, B., Chang, K., & Dong, A. (2005). Integrating Relevance Feedback Techniques for Image Retrieval Using Reinforcement Learning *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(10), 1536-1551.

Zhang, L., Lin, F., & Zhang, B. (2001). *Support Vector Machine Learning for Image Retrieval.* Paper presented at the International Conference on Image Processing.

Zhang, T., & Oles, F. (2000). *A Probability Analysis on The Value of Unlabeled Data for Classification Problems.* Paper presented at the Proceeding of 17th International Conference of Machine Learning, San Francisco, CA.

KEY TERMS

Heuristic Active Learning: The set of active learning algorithms in which the sample selection criteria is based on some heuristic objective function. For example, version space based active learning is to select the sample which can reduce the size of the version space.

Hypothesis Space: The set of all hypotheses in which the objective hypothesis is assumed to be found.

Semi-Supervised Learning: The set of learning algorithms in which both labelled and unlabelled data in the training dataset are directly used to train the classifier.

Statistical Active Learning: The set of active learning algorithms in which the sample selection criteria is based on some statistical objective function, such as minimization of generalisation error, bias and variance. Statistical active learning is usually statistically optimal.

Supervised Learning: The set of learning algorithms in which the samples in the training dataset are all labelled.

Unsupervised Learning: The set of learning algorithms in which the samples in training dataset are all unlabelled.

Version Space: The subset of the hypothesis space which is consistent with the training set.

Adaptive Algorithms for Intelligent Geometric Computing

M. L. Gavrilova
University of Calgary, Canada

INTRODUCTION

This chapter spans topics from such important areas as Artificial Intelligence, Computational Geometry and Biometric Technologies. The primary focus is on the proposed **Adaptive Computation Paradigm** and its applications to surface modeling and biometric processing.

Availability of much more affordable storage and high resolution image capturing devices have contributed significantly over the past few years to accumulating very large datasets of collected data (such as GIS maps, biometric samples, videos etc.). On the other hand, it also created significant challenges driven by the higher than ever volumes and the complexity of the data, that can no longer be resolved through acquisition of more memory, faster processors or optimization of existing algorithms. These developments justified the need for radically new concepts for massive data storage, processing and visualization. To address this need, the current chapter presents the original methodology based on the paradigm of the **Adaptive Geometric Computing**. The methodology enables storing complex data in a compact form, providing efficient access to it, preserving high level of details and visualizing dynamic changes in a smooth and continuous manner.

The first part of the chapter discusses adaptive algorithms in real-time visualization, specifically in GIS (Geographic Information Systems) applications. Data structures such as Real-time Optimally Adaptive Mesh (ROAM) and Progressive Mesh (PM) are briefly surveyed. The adaptive method **Adaptive Spatial Memory (ASM)**, developed by R. Apu and M. Gavrilova, is then introduced. This method allows fast and efficient visualization of complex data sets representing terrains, landscapes and Digital Elevation Models (DEM). Its advantages are briefly discussed.

The second part of the chapter presents application of adaptive computation paradigm and evolutionary computing to missile simulation. As a result, patterns of complex behavior can be developed and analyzed.

The final part of the chapter marries a concept of **adaptive computation** and **topology-based techniques** and discusses their application to challenging area of **biometric computing**.

BACKGROUND

For a long time, researchers were pressed with questions on how to model real-world objects (such as terrain, facial structure or particle system) realistically, while at the same time preserving rendering efficiency and space. As a solution, grid, mesh, TIN, Delaunay triangulation-based and other methods for model representation were developed over the last two decades. Most of these are static methods, not suitable for rendering dynamic scenes or preserving higher level of details.

In 1997, first methods for dynamic model representation: Real-time Optimally Adapting Mesh *(ROAM)* (Duchaineauy et. al., 1997, Lindstrom and Koller, 1996) and Progressive Mesh (PM) (Hoppe, 1997) were developed. Various methods have been proposed to reduce a fine mesh into an optimized representation so that the optimized mesh contains less primitives and yields maximum detail. However, this approach had two major limitations. Firstly, the cost of optimization is very expensive (several minutes to optimize one medium sized mesh). Secondly, the generated non-uniform mesh is still static. As a result, it yields poor quality when only a small part of the mesh is being observed. Thus, even with the further improvements, these methods were not capable of dealing with large amount of complex data or significantly varied level of details. They have soon were replaced by a different computational model for rendering geometric meshes (Li Sheng et. al. 2003, Shafae and Pajarola, 2003). The model employs a continuous refinement criteria based on an error metric to optimally adapt to a more accurate representation. Therefore, given a mesh representation and a small change in the viewpoint, the optimized mesh

for the next viewpoint can be computed by refining the existing mesh.

ADAPTIVE GEOMETRIC COMPUTING

This chapter presents *Adaptive Multi-Resolution Technique* for real-time terrain visualization utilizing a clever way of optimizing mesh dynamically for smooth and continuous visualization with a very high efficiency (frame rate) (Apu and Gavrilova (2005) (2007)). Our method is characterized by the efficient representation of massive underlying terrain, utilizes efficient transition between detail levels, and achieves frame rate constancy ensuring visual continuity. At the core of the method is *adaptive processing*: a formalized hierarchical representation that exploits the *subsequent refinement* principal. This allows us a full control over the complexity of the feature space. An error metric is assigned by a higher level process where objects (or features) are initially classified into different labels. Thus, this adaptive method is highly useful for feature space representation. In 2006, Gavrilova and Apu showed that such methods can act as a powerful tool not only for terrain rendering, but also for motion planning and adaptive simulations (Apu and Gavrilova, 2006). They introduced *Adaptive Spatial Memory (ASM)* model that utilizes adaptive approach for real-time online algorithm for multi-agent collaborative motion planning. They have demonstrate that the powerful notion of adaptive computation can be applied to perception and understanding of space. Extension of this method for 3D motion planning as part of collaborative research with Prof. I. Kolingerova group has been reported to be

significantly more efficient than conventional methods (Broz et.al., 2007).

We first move to discuss *evolutionary computing*. We demonstrate the power of adaptive computation by developing and applying adaptive computational model to missile simulation (Apu and Gavrilova, 2006). The developed adaptive algorithms described above have a property that spatial memory units can form, refine and collapse to simulate learning, adapting and responding to stimuli. The result is a complex multi-agent learning algorithm that clearly demonstrates organic behaviors such as sense of territory, trails, tracks etc. observed in flocks/herds of wild animals and insects. This gives a motivation to explore the mechanism in application to swarm behavior modeling.

Swarm Intelligence (SI) is the property of a system whereby the collective behaviors of unsophisticated agents interacting locally with their environment cause coherent functional global patterns to emerge (Bonabeau, 1999). Swarm intelligence provides a basis for exploration of a collective (distributed) behavior of a group of agents without centralized control or the provision of a global model. Agents in such system have limited perception (or intelligence) and cannot individually carry out the complex tasks. According to Bonebeau, by regulating the behavior of the agents in the swarm, one can demonstrate emergent behavior and intelligence as a collective phenomenon. Although the swarming phenomenon is largely observed in biological organisms such as an ant colony or a flock of birds, it is recently being used to simulate complex dynamic systems focused towards accomplishing a well-defined objective (Kennedy, 2001, Raupp ans Thalmann, 2001).

Figure 1. Split and merge operations in ASM model
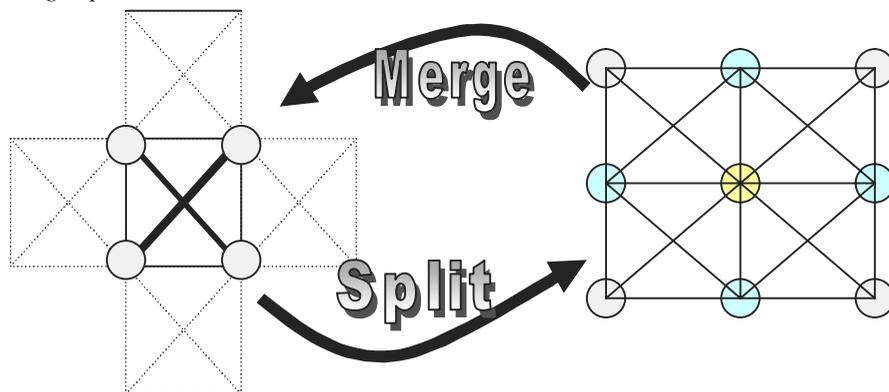
Let us now investigate application of the adaptive computational paradigm and swarm intelligence concept to missile behavior simulation (Apu and Gavrilova, 2006). First of all, let us note that complex strategic behavior can be observed by means of a task oriented artificial evolutionary process in which behaviors of individual missiles are described in surprising simplicity. Secondly, the global effectiveness and behavior of the missile swarm is relatively unaffected by disruption or destruction of individual units. From a strategic point of view, this adaptive behavior is a strongly desired property in military applications, which motivates our interest in applying it to missile simulation. Note that this problem was chosen as it presents a complex challenge for which an optimum solution is very hard to obtain using traditional methods. The dynamic and competitive relationship between missiles and turrets makes it extremely difficult to model using a deterministic approach. It should also be noted that the problem has an easy evaluation metric that allows determining fitness values precisely.

Now, let us summarize the idea of evolutionary optimization by applying genetic algorithm to evolve the missile genotype. We are particularly interested in observing the evolution of complex 3D formations and tactical strategies that the swarm learns to maximize their effectiveness during an attack simulation run. The simulation is based on attack, evasion and defense. While the missile sets strategy to strike the target, the battle ship prepares to shoot down as many missiles as possible (Figure 2 illustrates the basic missile ma-

Figure 2. Basic maneuvers for a missile using the Gene String
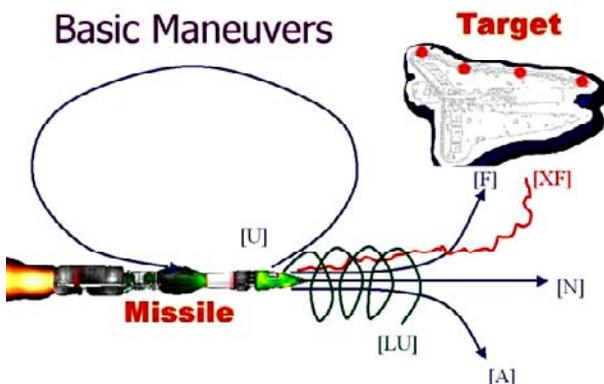
neuvers). Each attempt to destroy the target is called an attack simulation run. Its effectiveness equals to the number of missiles hitting the target. Therefore the outcome of the simulation is easily quantifiable. On the other hand, the interaction between missiles and the battleship is complex and nontrivial. As a result, war strategies may emerge in which a local penalty (i.e. sacrificing a missile) can optimize global efficiency (i.e. deception strategy). The simplest form of information known to each missile is its position and orientation and the location of the target. This information is augmented with information about missile neighborhood and environment, which influences missile navigation pattern. For actual missile behavior simulation, we use strategy based on the modified version of Boids flocking technique.

We have just outlined the necessary set of actions to reach the target or interact with the environment. This is the basic building block of missile navigation. The gene string is another important part that reflects the complexity with which such courses of action could be chosen. It contains a unique combination of maneuvers (such as attack, evasion, etc.) that evolve to create complex combined intelligence. We describe the fitness of the missile gene in terms of collective performance. After investigating various possibilities, we developed and used a two dimensional adaptive fitness function to evolve the missile strains in one evolutionary system. Details on this approach can be found in (Apu and Gavrilova, 2006).

After extensive experimentation, we have found many interesting characteristics, such as geometric attack formation and organic behaviors observed among swarms in addition to the highly anticipated strategies such as simultaneous attack, deception, retreat and other strategies (see Figure 3). We also examined the adaptability by randomizing the simulation coordinates, distance, initial formation, attack rate, and other parameters of missiles and measured the mean and variance of the fitness function. Results have shown that many of the genotypes that evolved are *highly adaptive* to the environment.

We have just reviewed the application of the adaptive computational paradigm to swarm intelligence and briefly described the efficient tactical swarm simulation method (Apu and Gavrilova 2006). The results clearly demonstrate that the swarm is able to develop complex strategy through the evolutionary process of genotype mutation. This contribution among other works on

adaptive computational intelligence will be profiled in detail in the upcoming book as part of Springer-Verlag book series on Computational Intelligence (Gavrilova, 2007).

As stated in the introduction, adaptive computation is based on a variable complexity level of detail paradigm, where a physical phenomenon can be simulated by the continuous process of local adaptation of spatial complexity. As presented by M. Gavrilova in Plenary Lecture at 3IA Eurographics Conference, France in 2006, the ***adaptive paradig*m** is a powerful computational model that can also be applied to vast area of biometric research. This section therefore reviews methods and techniques based on adaptive geometric methods in application to biometric problems. It emphasizes advantages that intelligent approach to geometric computing brings to the area of complex ***biometric data processing*** (Gavrilova 2007).

In information technology, ***biometric*s** refers to a study of physical and behavioral characteristics with the purpose of person identification (Yanushkevich, Gavrilova, Wang and Srihari, 2007). In recent years, the area of biometrics has witnessed a tremendous growth, partly as a result of a pressing need for increased security, and partly as a response to the new technological advances that are literally changing the way we live. Availability of much more affordable storage and the high resolution image biometric capturing devices have contributed to accumulating very large datasets of biometric data. In the earlier sections, we have studied the background of the adaptive mesh generation. Let us now look at the background research in topology-based data structures, and its application to biometric research. This information is highly relevant to goals of modeling and visualizing complex biometric data. At the same time as adaptive methodology was developing in GIS, interest to ***topology-based*** data structures, such as ***Voronoi diagrams*** *and* ***Delaunay triangulations***, has grown significantly. Some preliminary results on utilization of these topology-based data structures in biometric began to appear. For instance, research on image processing using Voronoi diagrams was presented in (Liang and Asano, 2004, Asano, 2006), studies of utilizing Voronoi diagram for fingerprint synthesis were conducted by (Bebis et. al., 1999, Capelli et. al. 2002), and various surveys of methods for modeling of human faces using triangular mesh appeared in (Wen and Huang, 2004, Li and Jain, 2005, Wayman et. al. 2005). Some interesting results were recently obtained in the BTLab, University of Calgary, through the development of topology-based feature extraction algorithms for fingerprint matching (Wang et. al. 2006, 2007, illustration is found in Figure 4), 3D facial expression modeling (Luo et. al. 2006) and iris synthesis (Wecker et. al. 2005). A comprehensive review of topology-based approaches in biometric modeling and synthesis can be found in recent book chapter on the subject (Gavrilova, 2007).

In this chapter, we propose to manage the challenges arising from large volumes of complex biometric data through the innovative utilization of the adaptive paradigm. We suggest combination of topology-based and hierarchy based methodology to store and search for biometric data, as well as to optimize such representation based on the data access and usage. Namely, retrieval of the data, or creating real-time visualization can be based on the dynamic patter of data usage (how often, what type of data, how much details, etc.), recorded and analyzed in the process of the biometric system being used for recognition and identification purposes.

Figure 3. Complex formation and attack patterns evolved

| *(a) Deception pattern* | *(b) Distraction pattern* | *(c) Organic motion pattern* |

Figure 4. Delaunay triangulation based technique for fingerprint matching
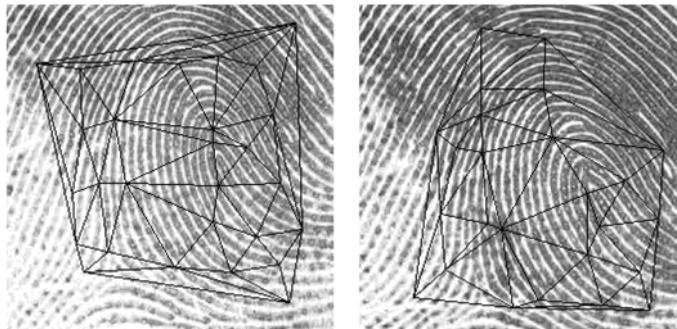
In addition to using this information for optimized data representation and retrieval, we also propose to incorporate intelligent learning techniques to predict most likely patters of the system usage and to represent and organize data accordingly.

On a practical side, to achieve our goal, we propose a novel way to represent complex biometric data through the organization of the data in a hierarchical tree-like structure. Such organization is similar in principle to the Adaptive Memory Subdivision (AMS), capable of representing and retrieving varies amount of information and level of detail that needs to be represented. Spatial quad-tree is used to hold the information about the system, as well as the instructions on how to process this information. Expansion is realized through the spatial subdivision technique that refines the data and increases level of details, and the collapsing is realized through the merge operation that simplifies the data representation and makes it more compact. The greedy strategy is used to *optimally adapt* to the best representation based on the user requirements, amount of available data and resources, required resolution and so on. This powerful technique enables us to achieve the goal of compact biometric data representation, that allows for instance to efficiently store minor details of the modeled face (e.g. scars, wrinkles) or detailed patterns of the iris.

FUTURE TRENDS

In addition to data representation, adaptive technique can be highly useful in biometric feature extraction with the purpose of fast and reliable retrieval and matching of the biometric data, and in implementing dynamic changes to the model. The methodology has a high potential of becoming one of the key approaches in biometric data modeling and synthesis.

CONCLUSION

The chapter reviewed the adaptive computational paradigm in application to surface modeling, evolutionary computing and biometric research. Some of the key future developments in the upcoming years will undoubtedly highlight the area, inspiring new generations of intelligent biometric systems with adaptive behavior.

REFERENCES

Apu R. & Gavrilova M (2005) Geo-Mass: Modeling Massive Terrain in Real-Time, *GEOMATICA J.* 59(3), 313-322.

Apu R. & Gavrilova M. (2006) Battle Swarm: An Evolutionary Approach to Complex Swarm Intelligence, *3IA Int. C. Comp. Graphics and AI,* Limoges, France, 139-150.

Apu, R & Gavrilova, M. (2007) Fast and Efficient Rendering System for Real-Time Terrain Visualization, *IJCSE Journal*, 2(2), 5/6.

Apu, R. & Gavrilova, M. (2006) An Efficient Swarm Neighborhood Management for a 3D Tactical Simulator, *IEEE-CS proceedings, ISVD 2006*, 85- 93

Asano, T. (2006) Aspect-Ratio Voronoi Diagram with Applications, *ISVD 2006, IEEE-CS proceedings*, 32-39

Bebis G., Deaconu T & Georiopoulous, M. (1999) Fingerprint Identification using Delaunay Triangulation, *ICIIS 99*, Maryland, 452-459

Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999) Swarm Intelligence: From Natural to Artificial Systems, NY: Oxford Univ. Press

Broz, P., Kolingerova, I, Zitka, P., Apu R. & Gavrilova M. (2007) Path planning in dynamic environment using an adaptive mesh, SCCG 2007, Spring Conference on Computer Graphics 2007, *ACM SIGGRAPH*

Capelli R, Maio, D, Maltoni D. (2002) Synthetic Fingerprint-Database Generation, *ICPR 2002*, Canada, vol 3, 369-376

Duchaineauy, M. et. al. (1997) ROAMing Terrain: Real-Time Optimally Adapting Meshes, *IEEE Visualization '97*, 81-88

Gavrilova M.L. (2007) Computational Geometry and Image Processing in Biometrics: on the Path to Convergence, in Book Image Pattern Recognition: Synthesis and Analysis in Biometrics, Book Chapter 4, 103-133, *World Scientific Publishers*

Gavrilova M.L. Computational Intelligence: A Geometry-Based Approach, in book series Studies in Computational Intelligence, *Springer-Verlag*, Ed. Janusz Kacprzyk, to appear.

Gavrilova, M.L. (2006) IEEE_CS Book of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering, *IEEE-CS*, Softcover, 2006, 270 pages.

Gavrilova, M.L. (2006) Geometric Algorithms in 3D Real-Time Rendering and Facial Expression Modeling, *3IA'2006 Plenary Lecture, Eurographics*, Limoges, France, 5-18

Hoppe, H. (1997) View-Dependent Refinement of Progressive Meshes, *SIGGRAPH '97 Proceedings,* 189-198

Kennedy, J., Eberhart, R. C., & Shi, Y. (2001) Swarm Intelligence*, San Francisco: Morgan Kaufmann Publishers*

Li Sheng, Liu Xuehui & Wu Enhau, (2003) Feature-Based Visibility-Driven CLOD for Terrain, *In Proc. Pacific Graphics 2003,* 313-322, IEEE Press

Li, S. & Jain, A. (2005) Handbook of Face Recognition. *Springer-Verlag*

Liang X.F. & Asano T. (2004) A fast denoising method for binary fingerprint image, *IASTED*, Spain, 309-313

Lindstrom, P. & Koller, D. (1996) Real-time continuous level of detail rendering of height fields*, SIGGRAPH 1996 Proceedings*, 109-118

Luo, Y, Gavrilova, M. & Sousa M.C. (2006) NPAR by Example: line drawing facial animation from photographs, CGIV'06, *IEEE, Computer Graphics, Imaging and Visualization***, 514-521

Raupp S. & Thalmann D. (2001) Hierarchical Model for Real Time Simulation of Virtual Human Crowds, *IEEE Trans. on Visualization and Computer Graphics* 7(2), 152-164

Shafae, M. & Pajarola, R. (2003) Dstrips: Dynamic Triangle Strips for Real-Time Mesh Simplification and Rendering, *Pacific Graphics 2003,* 271-280

Wang, C, Luo, Y, Gavrilova M & Rokne J. (2007) Fingerprint Image Matching Using a Hierarchical Approach, in Book Computational Intelligence in Information Assurance and Security, Springer SCI Series, 175-198

Wang, H, Gavrilova, M, Luo Y. & J. Rokne (2006) An Efficient Algorithm for Fingerprint Matching, *ICPR 2006, Int. C. on Pattern Recognition*, Hong Kong, IEEE-CS, 1034-1037

Wayman J, Jain A, Maltoni D & Maio D. (2005) Biometric Systems: Technology, Design and Performance Evaluation, Book, Springer

Wecker L**,** Samavati, F & Gavrilova M (2005) Iris Synthesis: A Multi-Resolution Approach, *GRAPHITE 2005, ACM Press*. 121-125

Wen, Z. & Huang, T. (2004) 3D Face Processing: Modeling, Analysis and Synthesis, *Kluwer*

Yanushkevich, S, Gavrilova M., Wang, P & Srihari S. (2007) Image Pattern Recognition: Synthesis and Analysis in Biometrics, Book World Scientific

KEY TERMS

Adaptive Geometric Model (AGM): A new approach to geometric computing utilizing adaptive computation paradigm. The model employs a continuous refinement criteria based on an error metric to optimally adapt to a more accurate representation.

Adaptive Multi-Resolution Technique (AMRT): For real-time terrain visualization is a method that utilizes a clever way of optimizing mesh dynamically for smooth and continuous visualization with a high efficiency.

Adaptive Spatial Memory (ASM): A hybrid method based on the combination of traditional hierarchical tree structure with the concept of expanding or collapsing tree nodes.

Biometric Technology (BT): An area of study of physical and behavioral characteristics with the purpose of person authentication and identification.

Delaunay Triangulation (DT): A computational geometry data structure dual to Voronoi diagram.

Evolutionary Paradigm (EP): The collective name for a number of problem solving methods utilizing principles of biological evolution, such as natural selection and genetic inheritance.

Swarm Intelligence (SI): The property of a system whereby the collective behaviors of unsophisticated agents interacting locally with their environment cause coherent functional global patterns to emerge.

Topology-Based Techniques (TBT): A group of methods using geometric properties of a set of objects in the space and their proximity

Voronoi Diagram (VD): A fundamental computational geometry data structure that stores topological information for a set of objects.

A

Adaptive Business Intelligence

Zbigniew Michalewicz
The University of Adelaide, Australia

INTRODUCTION

Since the computer age dawned on mankind, one of the most important areas in information technology has been that of "decision support." Today, this area is more important than ever. Working in dynamic and ever-changing environments, modern-day managers are responsible for an assortment of far reaching decisions: *Should the company increase or decrease its workforce? Enter new markets? Develop new products? Invest in research and development?* The list goes on. But despite the inherent complexity of these issues and the ever-increasing load of information that business managers must deal with, all these decisions boil down to two fundamental questions:

- What is likely to happen in the future?
- What is the best decision right now?

Whether we realize it or not, these two questions pervade our everyday lives — both on a personal and professional level. When driving to work, for instance, we have to make a traffic prediction before we can choose the quickest driving route. At work, we need to predict the demand for our product before we can decide how much to produce. And before investing in a foreign market, we need to predict future exchange rates and economic variables. It seems that regardless of the decision being made or its complexity, we first need to make a prediction of what is likely to happen in the future, and then make the best decision based on that prediction. This fundamental process underpins the basic premise of *Adaptive Business Intelligence*.

BACKGROUND

Simply put, Adaptive Business Intelligence is the discipline of combining prediction, optimization, and adaptability into a system capable of answering these two fundamental questions: *What is likely to happen in the future?* and *What is the best decision right now?*

(Michalewicz et al. 2007). To build such a system, we first need to understand the methods and techniques that enable prediction, optimization, and adaptability (Dhar and Stein, 1997). At first blush, this subject matter is nothing new, as hundreds of books and articles have already been written on business intelligence (Vitt et al., 2002; Loshin, 2003), data mining and prediction methods (Weiss and Indurkhya, 1998; Witten and Frank, 2005), forecasting methods (Makridakis et al., 1988), optimization techniques (Deb 2001; Coello et al. 2002; Michalewicz and Fogel, 2004), and so forth. However, none of these has explained how to combine these various technologies into a software system that is capable of predicting, optimizing, and adapting. *Adaptive Business Intelligence* addresses this very issue.

Clearly, the future of the business intelligence industry lies in systems that can make decisions, rather than tools that produce detailed reports (Loshin 2003). As most business managers now realize, there is a world of difference between having good knowledge and detailed reports, and making smart decisions. Michael Kahn, a technology reporter for Reuters in San Francisco, makes a valid point in the January 16, 2006 story entitled "Business intelligence software looks to future":

"But analysts say applications that actually answer questions rather than just present mounds of data is the key driver of a market set to grow 10 per cent in 2006 or about twice the rate of the business software industry in general.

'Increasingly you are seeing applications being developed that will result in some sort of action,' said Brendan Barnacle, an analyst at Pacific Crest Equities. 'It is a relatively small part now, but it is clearly where the future is. That is the next stage of business intelligence.'"

MAIN FOCUS OF THE CHAPTER

"The answer to my problem is hidden in my data ... but I cannot dig it up!" This popular statement has been around for years as business managers gathered and stored massive amounts of data in the belief that they contain some valuable insight. But business managers eventually discovered that raw data are rarely of any benefit, and that their real value depends on an organization's ability to analyze them. Hence, the need emerged for software systems capable of retrieving, summarizing, and interpreting data for end-users (Moss and Atre, 2003).

This need fueled the emergence of hundreds of *business intelligence* companies that specialized in providing software systems and services for extracting *knowledge* from raw data. These software systems would analyze a company's operational data and provide knowledge in the form of tables, graphs, pies, charts, and other statistics. For example, a business intelligence report may state that 57% of customers are between the ages of 40 and 50, or that product X sells much better in Florida than in Georgia.[1]

Consequently, the general goal of most business intelligence systems was to: (1) access data from a variety of different sources; (2) transform these data into information, and then into knowledge; and (3) provide an easy-to-use graphical interface to display this knowledge. In other words, a business intelligence system was responsible for collecting and digesting data, and presenting knowledge in a friendly way (thus enhancing the end-user's ability to make good decisions). The diagram in Figure 1 illustrates the processes that underpin a traditional business intelligence system.

Although different texts have illustrated the relationship between data and knowledge in different ways (e.g.,

Davenport and Prusak, 2006; Prusak, 1997; Shortliffe and Cimino, 2006), the commonly accepted distinction between data, information, and knowledge is:

- *Data* are collected on a daily basis in the form of bits, numbers, symbols, and "objects."
- *Information* is "organized data," which are pre-processed, cleaned, arranged into structures, and stripped of redundancy.
- *Knowledge* is "integrated information," which includes facts and relationships that have been perceived, discovered, or learned.

Because knowledge is such an essential component of any decision-making process (as the old saying goes, *"Knowledge is power!")*, many businesses have viewed knowledge as the final objective. But it seems that knowledge is no longer enough. A business may "know" a lot about its customers — it may have hundreds of charts and graphs that organize its customers by age, preferences, geographical location, and sales history — but management may still be unsure of what decision to make! And here lies the difference between "decision support" and "decision making": all the knowledge in the world will not guarantee the right or best decision.

Moreover, recent research in psychology indicates that widely held beliefs can actually hamper the decision-making process. For example, common beliefs like "the more knowledge we have, the better our decisions will be," or "we can distinguish between useful and irrelevant knowledge," are not supported by empirical evidence. Having more knowledge merely increases our confidence, but it does not improve the accuracy of our decisions. Similarly, people supplied with "good" and "bad" knowledge often have trouble distinguishing

Figure 1. The processes that underpin a traditional business intelligence system
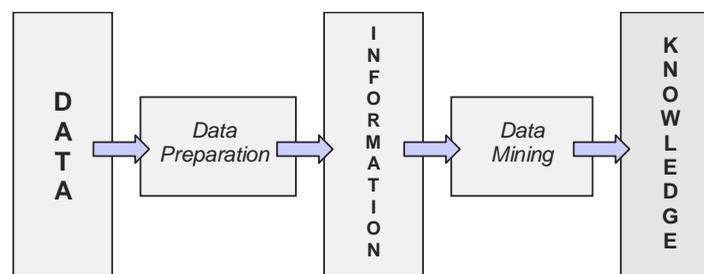
between the two, proving that irrelevant knowledge decreases our decision-making effectiveness.

Today, most business managers realize that a gap exists between having the right knowledge and making the right decision. Because this gap affects management's ability to answer fundamental business questions (such as "What should be done to increase profits? Reduce costs? Or increase market share?"), the future of business intelligence lies in systems that can provide answers and recommendations, rather than mounds of knowledge in the form of reports. *The future of business intelligence lies in systems that can make decisions!* As a result, there is a new trend emerging in the marketplace called *Adaptive Business Intelligence.* In addition to performing the role of traditional business intelligence (transforming data into knowledge), Adaptive Business Intelligence also includes the decision-making process, which is based on prediction and optimization as shown in Figure 2.

While *business intelligence* is often defined as "a broad category of application programs and technologies for gathering, storing, analyzing, and providing access to data," the term *Adaptive Business Intelligence* can be defined as "the discipline of using prediction and optimization techniques to build self-learning 'decisioning' systems" (as the above diagram shows). Adaptive Business Intelligence systems include elements of data mining, predictive modeling, forecasting, optimization, and adaptability, and are used by business managers to make better decisions.

This relatively new approach to business intelligence is capable of recommending the best course of action (based on past data), but it does so in a very special way: An Adaptive Business Intelligence system incorporates prediction and optimization modules to recommend near-optimal decisions, and an "adaptability module" for improving future recommendations. Such systems can help business managers make decisions that increase efficiency, productivity, and competitiveness. Furthermore, the importance of *adaptability* cannot be overemphasized. After all, what is the point of using a software system that produces sub par schedules, inaccurate demand forecasts, and inferior logistic plans, time after time? Would it not be wonderful to use a software system that could *adapt* to changes in the marketplace? A software system that could *improve* with time?

FUTURE TRENDS

The concept of adaptability is certainly gaining popularity, and not just in the software sector. Adaptability has already been introduced in everything from automatic car transmissions (which adapt their gear-change patterns to a driver's driving style), to running shoes (which adapt their cushioning level to a runner's size and stride), to Internet search engines (which adapt their search results to a user's preferences and prior search history). These products are very appealing for individual consumers, because, despite their mass production, they are capable of adapting to the preferences of each unique owner after some period of time.

The growing popularity of adaptability is also underscored by a recent publication of the US De-

Figure 2. Adaptive business intelligence system
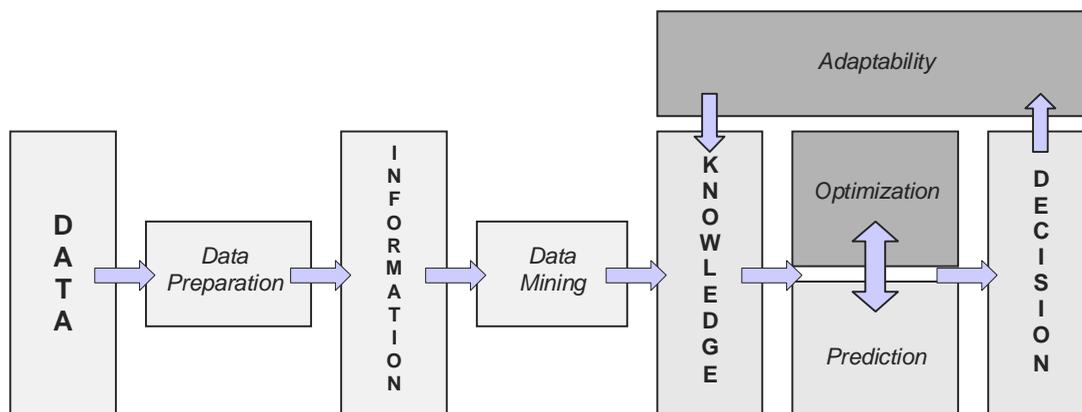
partment of Defense. This lists 19 important research topics for the next decade and many of them include the term "adaptive": *Adaptive* Coordinated Control in the Multi-agent 3D Dynamic Battlefield, Control for *Adaptive* and Cooperative Systems, *Adaptive* System Interoperability, *Adaptive* Materials for Energy-Absorbing Structures, and Complex *Adaptive* Networks for Cooperative Control.

For sure, adaptability was recognized as important component of intelligence quite some time ago: Alfred Binet (born 1857), French psychologist and inventor of the first usable intelligence test, defined intelligence as "... judgment, otherwise called good sense, practical sense, initiative, the faculty of *adapting* one's self to circumstances." Adaptability is a vital component of any intelligent system, as it is hard to argue that a system is "intelligent" if it does not have the capacity to adapt. For humans, the importance of adaptability is obvious: our ability to adapt was a key element in the evolutionary process. In psychology, a behavior or trait is adaptive when it helps an individual adjust and function well within a changing social environment. In the case of artificial intelligence, consider a chess program capable of beating the world chess master: Should we call this program intelligent? Probably not. We can attribute the program's performance to its ability to evaluate the current board situation against a multitude of possible "future boards" before selecting the best move. However, because the program cannot learn or adapt to new rules, the program will lose its effectiveness if the rules of the game are changed or modified. Consequently, because the program is incapable of learning or adapting to new rules, the program is not intelligent.

The same holds true for any expert system. No one questions the usefulness of expert systems in some environments (which are usually well defined and static), but expert systems that are incapable of learning and adapting should not be called "intelligent." Some expert knowledge was programmed in, that is all.

So, what are the future trends for Adaptive Business Intelligence? In words of Jim Goodnight, the CEO of SAS Institute (Collins et al. 2007):

"Until recently, business intelligence was limited to basic query and reporting, and it never really provided that much intelligence ...."

However, this is about to change. Keith Collins, the Chief Technology Officer of SAS Institute (Collins et al. 2007) believes that:

"A new platform definition is emerging for business intelligence, where BI is no longer defined as simple query and reporting. [...] In the next five years, we'll also see a shift in performance management to what we're calling predictive performance management, where analytics play a huge role in moving us beyond just simple metrics to more powerful measures."

Further, Jim Davis, the VP Marketing of SAS Institute (Collins et al. 2007) stated:

"In the next three to five years, we'll reach a tipping point where more organizations will be using BI to focus on how to optimize processes and influence the bottom line ...."

Finally, it would be important to incorporate *adaptability* in prediction and optimization components of the future Adaptive Business Intelligence systems.

There are some recent, successful implementations of Adaptive Business Intelligence systems reported (e.g., Michalewicz et al. 2005), which provide daily decision support for large corporations and result in multi-million dollars return on investment. There are also companies (e.g., www.solveitsoftware.com) which specialize in development of Adaptive Business Intelligence tools. However, further research effort is required. For example, most of the research in machine learning has focused on using historical data to build prediction models. Once the model is built and evaluated, the goal is accomplished. However, because new data arrive at regular intervals, building and evaluating a model is just the first step in Adaptive Business Intelligence. Because these models need to be updated regularly (something that the adaptability module is responsible for), we expect to see more emphasis on this updating process in machine learning research. Also, the frequency of updating the prediction module, which can vary from seconds (e.g., in real-time currency trading systems), to weeks and months (e.g., in fraud detection systems) may require different techniques and methodologies. In general, Adaptive Business Intelligence systems would include the research results from control theory, statistics, operations research, machine learning, and modern heuristic methods, to name a few. We also

expect that major advances will continue to be made in modern optimization techniques. In the years to come, more and more research papers will be published on constrained and multi-objective optimization problems, and on optimization problems set in dynamic environments. This is essential, as most real-world business problems are constrained, multi-objective, and set in a time-changing environment.

Intelligence is all about. Systems based on Adaptive Business Intelligence aim at solving real-world business problems that have complex constraints, are set in time-changing environments, have several (possibly conflicting) objectives, and where the number of possible solutions is too large to enumerate. Solving these problems requires a system that incorporates modules for prediction, optimization, and adaptability.

CONCLUSION

It is not surprising that the fundamental components of Adaptive Business Intelligence are already emerging in other areas of business. For example, the *Six Sigma* methodology is a great example of a well-structured, data-driven methodology for eliminating defects, waste, and quality-control problems in many industries. This methodology recommends the sequence of steps shown in Figure 3.

Note that the above sequence is very close "in spirit" to part of the previous diagram, as it describes (in more detail) the adaptability control loop. Clearly, we have to "measure," "analyze," and "improve," as we operate in a dynamic environment, so the process of improvement is continuous. The SAS Institute proposes another methodology, which is more oriented towards data mining activities. Their methodology recommends the sequence of steps shown in Figure 4.

Again, note that the above sequence is very close to another part of our diagram, as it describes (in more detail) the transformation from data to knowledge. It is not surprising that businesses are placing considerable emphasis on these areas, because better decisions usually translate into better financial performance. And better financial performance is what Adaptive Business

REFERENCES

Coello, C.A.C., Van Veldhuizen, A.A., and Lamont, G.B. (2002). *Evolutionary algorithms for solving multi-objective problems.* Kluwer Academic.

Collins, K., Goodnight, J., Hagström, M., Davis, J. (2007). The future of business intelligence: Four questions, four views. S*ASCOM, First quarter,* 2007.

Davenport, T.H. and Prusak, L. (2006). *Working knowledge.* Academic Internet Publishers.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms.*Wiley.

Dhar, V. and Stein, R., (1997). *Seven methods for transforming corporate data into business intelligence.* Prentice Hall.

Loshin, D. (2003). *Business intelligence: The savvy manager's guide.* Margan Kaufmann.

Makridakis, S., Wheelwright, S.C., and Hyndman, R.J. (1998). *Forecasting: Methods and applications.* Wiley.

Michalewicz, Z. and Fogel, D.B. (2004). *How to solve it: Modern heuristics,* 2nd edition. Springer.

Figure 3. Six Sigma methodology sequence

Define	Measure	Analyze	Improve	Control
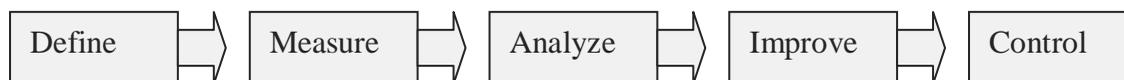
Figure 4. SAS Institute recommended methodolgy sequence

Sample	Explore	Modify	Model	Assess

Michalewicz, Z., Schmidt, M., Michalewicz, M., and Chiriac, C. (2005). A decision-support system based on computational intelligence: A case study. *IEEE Intelligent Systems, 20*(4), 44-49.

Michalewicz, Z., Schmidt, M., Michalewicz, M., and Chiriac, C. (2007). *Adaptive business intelligence.* Springer.

Moss, L. T. and Atre, S. (2003). *Business intelligence roadmap.* Addison Wesley.

Prusak, L. (1997). *Knowledge in organizations.* Butterworth-Heinemann.

Shortliffe, E. H. and Cimino, J. J. Eds (2006). *Biomedical informatics: Computer applications in health care and biomedicine.* Springer.

Vitt, E., Luckevich, M., and Misner, S. (2002). *Business intelligence: Making better decisions faster.* Microsoft Press.

Weiss, S. M. and Indurkhya, N., (1998). *Predictive data mining.* Morgan Kaufmann.

Witten, I. H. and Frank, E. (2005). *Data mining: Practical machine learning tools and techniques,* 2nd edition. Morgan Kaufmann.

TERMS AND DEFINITIONS

A

Adaptive Business Intelligence: The discipline of using prediction and optimization techniques to build self-learning 'decisioning' systems''.

Business Intelligence: A collection of tools, methods, technologies, and processes needed to transform data into actionable knowledge.

Data: Pieces collected on a daily basis in the form of bits, numbers, symbols, and "objects."

Data Mining: The application of analytical methods and tools to data for the purpose of identifying patterns, relationships, or obtaining systems that perform useful tasks such as classification, prediction, estimation, or affinity grouping.

Information: "Organized data," which are preprocessed, cleaned, arranged into structures, and stripped of redundancy.

Knowledge: "Integrated information," which includes facts and relationships that have been perceived, discovered, or learned.

Optimization: Process of finding the solution that is the best fit to the available resources.

Prediction: A statement or claim that a particular event will occur in the future.

ENDNOTE

[1]   Note that *business intelligence* can be defined both as a "state" (a report that contains knowledge) and a "process" (software responsible for converting data into knowledge).

Adaptive Neural Algorithms for PCA and ICA

Radu Mutihac
University of Bucharest, Romania

INTRODUCTION

Artificial neural networks (ANNs) (McCulloch & Pitts, 1943) (Haykin, 1999) were developed as models of their biological counterparts aiming to emulate the real neural systems and mimic the structural organization and function of the human brain. Their applications were based on the ability of self-designing to solve a problem by learning the solution from data. A comparative study of neural implementations running principal component analysis (PCA) and independent component analysis (ICA) was carried out. Artificially generated data additively corrupted with white noise in order to enforce randomness were employed to critically evaluate and assess the reliability of data projections. Analysis in both time and frequency domains showed the superiority of the estimated independent components (ICs) relative to principal components (PCs) in faithful retrieval of the genuine (latent) source signals.

Neural computation belongs to information processing dealing with adaptive, parallel, and distributed (localized) signal processing. In data analysis, a common task consists in finding an adequate subspace of multivariate data for subsequent processing and interpretation. Linear transforms are frequently employed in data model selection due to their computational and conceptual simplicity. Some common linear transforms are PCA, factor analysis (FA), projection pursuit (PP), and, more recently, ICA (Comon, 1994). The latter emerged as an extension of nonlinear PCA (Hotelling, 1993) and developed in the context of blind source separation (BSS) (Cardoso, 1998) in signal and array processing. ICA is also related to recent theories of the visual brain (Barlow, 1991), which assume that consecutive processing steps lead to a progressive reduction in the redundancy of representation (Olshausen and Field, 1996).

This contribution is an overview of the PCA and ICA neuromorphic architectures and their associated algorithmic implementations increasingly used as exploratory techniques. The discussion is conducted on artificially generated sub- and super-Gaussian source signals.

BACKGROUND

In neural computation, transforming methods amount to unsupervised learning, since the representation is only learned from data without any external control. Irrespective of the nature of learning, the neural adaptation may be formally conceived as an optimization problem: an objective function describes the task to be performed by the network and a numerical optimization procedure allows adapting network parameters (e.g., connection weights, biases, internal parameters). This process amounts to search or nonlinear programming in a quite large parameter space. However, any prior knowledge available on the solution might be efficiently exploited to narrow the search space. In supervised learning, the additional knowledge is incorporated in the net architecture or learning rules (Gold, 1996). A less extensive research was focused on unsupervised learning. In this respect, the mathematical methods usually employed are drawn from classical constrained multivariate nonlinear optimization and rely on the Lagrange multipliers method, the penalty or barrier techniques, and the classical numerical algebra techniques, such as deflation/renormalization (Fiori, 2000), the Gram-Schmidt orthogonalization procedure, or the projection over the orthogonal group (Yang, 1995).

PCA and ICA Models

Mathematically, the linear stationary PCA and ICA models can be defined on the basis of a common data model. Suppose that some stochastic processes are represented by three random (column) vectors $\mathbf{x}(t)$, $\mathbf{n}(t) \in \mathbb{R}^N$ and $\mathbf{s}(t) \in \mathbb{R}^M$ with zero mean and finite covariance, with the components of $\mathbf{s}(t) = \{s_1(t), s_2(t), ..., s_M(t)\}$ being statistically independent and at most one Gaussian. Let $\mathbf{A}$ be a rectangular constant full column rank $N \times M$ matrix with at least as many rows as columns ($N \geq M$), and denote by t the sample index (i.e., time or sample point) taking the discrete values $t = 1, 2, ...,$

T. We postulate the existence of a linear relationship among these variables like:

$$\mathbf{x}(t) = \mathbf{As}(t) + \mathbf{n}(t) = \sum_{i=1}^{M} s_i(t)\,\mathbf{a}_i + \mathbf{n}(t) \quad (1)$$

Here $\mathbf{s}(t)$, $\mathbf{x}(t)$, $\mathbf{n}(t)$, and $\mathbf{A}$ are the sources, the observed data, the (unknown) noise in data, and the (unknown) mixing matrix, respectively, whereas $\mathbf{a}_i$, $i = 1, 2, ..., M$ are the columns of $\mathbf{A}$. Mixing is supposed to be instantaneous, so there is no time delay between a (latent) source variable $s_i(t)$ mixing into an observable (data) variable $x_j(t)$, with $i = 1, 2, ..., M$ and $j = 1, 2, ..., N$.

Consider that the stochastic vector process $\{\mathbf{x}(t)\} \in \mathbb{R}^N$ has the mean $E\{\mathbf{x}(t)\} = 0$ and the covariance matrix $\mathbf{C_x} = E\{\mathbf{x}(t)\,\mathbf{x}(t)^T\}$. The goal of PCA is to identify the dependence structure in each dimension and to come out with an orthogonal transform matrix $\mathbf{W}$ of size $L \times N$ from $\mathbb{R}^N$ to $\mathbb{R}^L$, $L < N$, such that the L-dimensional output vector $\mathbf{y}(t) = \mathbf{W}\,\mathbf{x}(t)$ sufficiently represents the intrinsic features of the input data, and where the covariance matrix $\mathbf{C_y}$ of $\{\mathbf{y}(t)\}$ is a diagonal matrix $\mathbf{D}$ with the diagonal elements arranged in descending order, $d_{i,i} \geq d_{i+1,i+1}$. The restoration of $\{\mathbf{x}(t)\}$ from $\{\mathbf{y}(t)\}$, say $\{\hat{\mathbf{x}}(t)\}$, is consequently given by $\hat{\mathbf{x}}(t) = \mathbf{W}^T \mathbf{W}\,\mathbf{x}(t)$ (Figure 1). For a given L, PCA aims to find an optimal value of $\mathbf{W}$, such as to minimize the error function $J = E\{\|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\|\}$. The rows in $\mathbf{W}$ are the PCs of the stochastic process $\{\mathbf{x}(t)\}$ and the eigenvectors $\mathbf{c}_j$, $j = 1, 2, ..., L$ of the input covariance matrix $\mathbf{C_x}$. The subspace spanned by the principal eigenvectors $\{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_L\}$ with $L < N$, is called the PCA subspace of dimensionality L.

The ICA problem can be formulated as following: given T realizations of $\mathbf{x}(t)$, estimate both the matrix $\mathbf{A}$ and the corresponding realizations of $\mathbf{s}(t)$. In BSS the task is somewhat relaxed to finding the waveforms $\{s_i(t)\}$ of the sources knowing only the (observed) mixtures $\{x_j(t)\}$. If no suppositions are made about the noise, the additive noise term is omitted in (1). A practical strategy is to include noise in the signals as supplementary term(s): hence the ICA model (Fig. 2) becomes:

$$\mathbf{x}(t) = \mathbf{As}(t) = \sum_{i=1}^{M} \mathbf{a}_i s_i(t) \quad (2)$$

The source separation consists in updating an unmixing matrix $\mathbf{B}(t)$, without resorting to any information about the spatial mixing matrix $\mathbf{A}$, so that the output vector $\mathbf{y}(t) = \mathbf{B}(t)\,\mathbf{x}(t)$ becomes an estimate $\mathbf{y}(t) = \hat{\mathbf{s}}(t)$ of the original independent source signals $\mathbf{s}(t)$. The separating matrix $\mathbf{B}(t)$ is divided in two parts dealing with dependencies in the first two moments, i.e., the whitening matrix $\mathbf{V}(t)$, and the dependencies in

Figure 1. Schematic of the PCA model
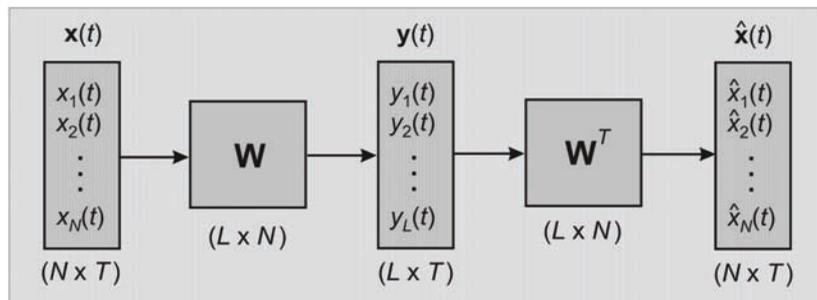
Figure 2. Schematic of the ICA model
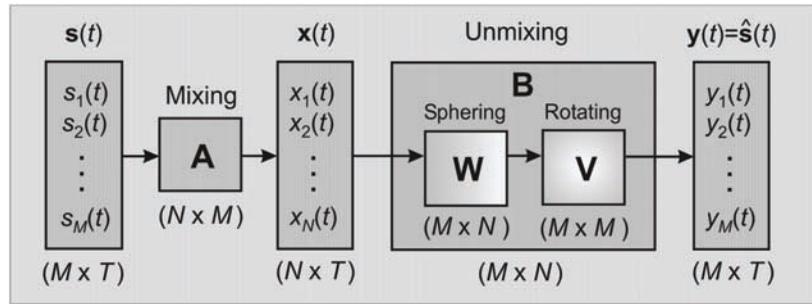
Figure 3. A simple feed-forward ANN performing PCA and ICA
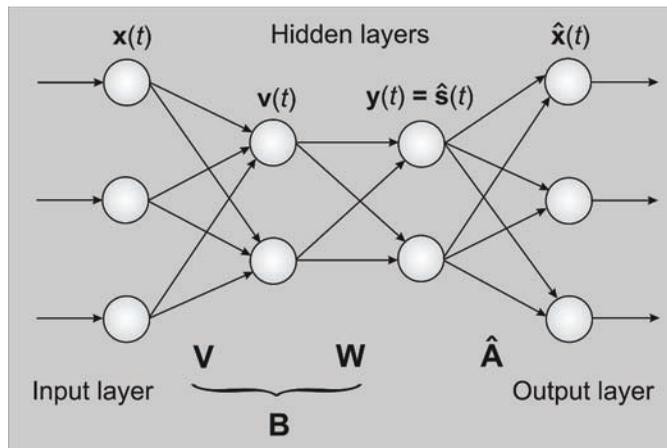
higher-order statistics, i.e., the orthogonal separating matrix $\mathbf{W}(t)$ in the whitened space (Fig. 2). If we assume zero-mean observed data $\mathbf{x}(t)$, then we get by whitening a vector $\mathbf{v}(t) = \mathbf{V}(t)\,\mathbf{x}(t)$ with decorrelated components. The subsequent linear transform $\mathbf{W}(t)$ seeks the solution by an adequate rotation in the space of component densities and yields $\mathbf{y}(t) = \mathbf{W}(t)\,\mathbf{v}(t)$ (Fig. 2). The total separation matrix between the input and the output layer turns to be $\mathbf{B}(t) = \mathbf{W}(t)\,\mathbf{V}(t)$. In the standard stationary case, the whitening and the orthogonal separating matrices converge to some constant values after a finite number of iterations during learning, that is, $\mathbf{B}(t) \to \mathbf{B} = \mathbf{W}\,\mathbf{V}$.

NEURAL IMPLEMENTATIONS

A neural approach to BSS entails a network that has mixtures of the source signals as input and produces approximations of the source signals as output (Figure 3). As a prerequisite, the input signals must be mutually uncorrelated, a requirement usually fulfilled by PCA. The output signals must nevertheless be mutually independent, which leads in a natural way from PCA to ICA. The higher order statistics required by source separation can be incorporated into computations either explicitly or by using suitable nonlinearities. ANNs better fit the latter approach (Karhunen, 1996).

The core of the large class of neural adaptive algorithms consists in a learning rule and its associated optimization criterion (objective function). These two items differentiate the algorithms, which are actually families of algorithms parameterized by the nonlinear

function used. An update rule is specified by the iterative incremental change $\Delta\mathbf{W}$ of the rotation matrix $\mathbf{W}$, which gives the general form of the learning rule:

$$\mathbf{W} \to \mathbf{W} + \Delta\mathbf{W} \qquad (3)$$

Neural PCA

First, consider a single artificial neuron receiving an M-dimensional input vector $\mathbf{x}$. It gradually adapts its weight vector $\mathbf{w}$ so that the function $E\left\{f\left(\mathbf{w}^T\mathbf{x}\right)\right\}$ is maximized, where E is the expectation with respect to the (unknown) probability density of $\mathbf{x}$ and f is a continuous objective function. The function f is bounded by setting constant the Euclidian norm of $\mathbf{w}$. A constrained gradient ascent learning rule based on a sequence of sample functions for relatively small learning rates $\alpha\left(t\right)$ is then (Oja, 1995):

$$\mathbf{w}\left(t+1\right)= \mathbf{w}\left(t\right)+a\left(t\right)\left(\mathbf{I}-\mathbf{w}\left(t\right)^T\mathbf{w}\left(t\right)\right)\mathbf{x}\left(t\right) g\left(\mathbf{w}\left(t\right)^T\mathbf{w}\left(t\right)\right)$$

$$(4)$$

where $g = f'$. Any PCA learning rules tend to find that direction in the input space along which the data has maximal variance. If all directions in the input space have equal variance, the one-unit case with a suitable nonlinearity is approximately minimizing the kurtosis of the neuron input. It means that the weight vector of the unit will be determined by the direction in the input space on which the projection of the input data is mostly clustered and deviates significantly from normality. This task is essentially the goal in the PP technique.

In the case of single layer ANNs consisting of L parallel units, with each unit i having the same M-element input vector $\mathbf{x}$ and its own weight vector $\mathbf{w}_i$ that together comprise an $M \times L$ weight matrix $\mathbf{W} = \left[\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_L\right]$ the following training rule obtained from (4) is a generalization of the linear PCA learning rule (in matrix form):

$$\mathbf{W}\left(t+1\right)= \mathbf{W}\left(t\right)+a\left(t\right)\left(\mathbf{I}-\mathbf{W}\left(t\right)\mathbf{W}\left(t\right)^T\right)\mathbf{x}\left(t\right) g\left(\mathbf{x}\left(t\right)^T\mathbf{W}\left(t\right)\right)$$

$$(5)$$

Due to the instability of the above nonlinear Hebbian learning rule for the multi-unit case, a different approach based on optimizing two criteria simultaneously was introduced (Oja, 1982):

$$\mathbf{W}\left(t+1\right)= \mathbf{W}\left(t\right)+\mu\left(t\right)\mathbf{x}\left(t\right) g\left(\mathbf{y}\left(t\right)^T\right) +\gamma\left(t\right)\left(\mathbf{I}-\mathbf{W}\left(t\right)\mathbf{W}\left(t\right)^T\right)$$

$$(6)$$

Here $\mu\left(t\right)$ is chosen positive or negative depending on our interest in maximizing or minimizing, respectively, the objective function $J_1\left(\mathbf{w}_i\right)= E\left\{f\left(\mathbf{x}^T\mathbf{w}_i\right)\right\}$. Similarly, $\gamma\left(t\right)$ is another gain parameter that is always positive and constrains the weight vectors to orthonormality, which is imposed by an appropriate penalty function such as:

$$J_2\left(\mathbf{w}_i\right)= \frac{1}{2}\left(1-\mathbf{w}_i^T\mathbf{w}_i\right)^2 + \frac{1}{2}\sum_{j=1, j\neq i}^{M}\left(\mathbf{w}_i^T\mathbf{w}_j\right)^2.$$

This is the bigradient algorithm, which is iterated until the weight vectors have converged with the desired accuracy. This algorithm can use normalized Hebbian or anti-Hebbian learning in a unified formula. Starting from one-unit rule, the multi-unit bigradient algorithm can simultaneously extract several robust counterparts of the principal or minor eigenvectors of the data covariance matrix (Wang, 1996).

In the case of multilayered ANNs, the transfer functions of the hidden nodes can be expressed by radial basis functions (RBF), whose parameters could be learnt by a two-stage gradient descent strategy. A new growing RBF-node insertion strategy with different RBF is used in order to improve the net performances. The learning strategy is reported to save computational time and memory space in approximation of continuous and discontinuous mappings (Esposito *et al.*, 2000).

Neural ICA

Various forms of unsupervised learning have been implemented in ANNs beyond standard PCA like nonlinear PCA and ICA. Data whitening can be neurally emulated by PCA with a simple iterative algorithm that updates the sphering matrix $\mathbf{V}\left(t\right)$:

A

$$\mathbf{V}(t+1) = \mathbf{V}(t) - \alpha(t)(\mathbf{vv}^T - \mathbf{I}) \qquad (7)$$

After getting the decorrelation matrix $\mathbf{V}(t)$, the basic task for ICA algorithms remains to come out with an orthogonal matrix $\mathbf{W}(t)$, which is equivalent to a suitable rotation of the decorrelated data $\mathbf{v}(t) = \mathbf{V}(t)\mathbf{x}(t)$ aiming to maximize the product of the marginal densities of its components. There are various neural approaches to estimate the rotation matrix $\mathbf{W}(t)$. An important class of algorithms is based on maximization of network entropy (Bell, 1995). The BS nonlinear information maximization (infomax) algorithm performs online stochastic gradient ascent in mutual information (MI) between outputs and inputs of a network. By minimizing the MI between outputs, the network factorizes the inputs into independent components. Considering a network with the input vector $\mathbf{x}(t)$, a weight matrix $\mathbf{W}(t)$, and a monotonically transformed output vector $\mathbf{y} = g(\mathbf{Wx} + \mathbf{w}_0)$, then the resulting learning rule for the weights and bias-weights, respectively, are:

$$\Delta \mathbf{W} = \left[\mathbf{W}^T\right]^{-1} + \mathbf{x}(\mathbf{1} - 2\mathbf{y})^T \quad \text{and} \quad \Delta \mathbf{w}_0 = \mathbf{1} - 2\mathbf{y}$$

$$(8)$$

In the case of bounded variables, the interplay between the anti-Hebbian term $\mathbf{x}(\mathbf{1} - 2\mathbf{y})^T$ and the antidecay term $\left[\mathbf{W}^T\right]^{-1}$ produces an output density that is close to the flat constant distribution, which corresponds to the maximum entropy distribution. Amari, Cichocki, and Yang (Amari, 1996) altered the BS infomax algorithm by using the natural gradient instead of the stochastic gradient to reduce the complexity of neural computations and significantly improving the speed of convergence. The update rule proposed for the separating matrix is:

$$\Delta \mathbf{W} = \left[\mathbf{I} - g(\mathbf{Wx})(\mathbf{Wx})^T\right]\mathbf{W} \qquad (9)$$

Lee *et al.* (Lee, 2000) extended to both sub-and super-Gaussian distributions the learning rule developed from the infomax principle satisfying a general stability criterion and preserving the simple initial architecture of the network. Applying either natural or relative gradient (Cardoso, 1996) for optimization, their learning rule yields results that compete with fixed-point batch computations.

The equivariant adaptive separation via independence (EASI) algorithm introduced by Cardoso and Laheld (1996) is a nonlinear decorrelation method. The objective function $J(\mathbf{W}) = E\{f(\mathbf{Wx})\}$ is subject to minimization with the orthogonal constraint imposed on $\mathbf{W}$ and the nonlinearity $g = f'$ chosen according to data kurtosis. Its basic update rule equates to:

$$\Delta \mathbf{W} = -\lambda \left(\mathbf{yy}^T - \mathbf{I} + g(\mathbf{y})\mathbf{y}^T - \mathbf{y}g(\mathbf{y}^T)\right)\mathbf{W}$$

$$(10)$$

Fixed-point (FP) algorithms are searching the ICA solution by minimizing mutual information (MI) among the estimated components (Hyvärinen, 1997). The FastICA learning rule finds a direction $\mathbf{w}$ so that the projection of $\mathbf{w}^T\mathbf{x}$ maximizes a contrast function of the form $J_G(\mathbf{w}) = \left[E\{f(\mathbf{w}^T\mathbf{x})\} - E\{f(\mathbf{v})\}\right]^2$ with $\mathbf{v}$ standing for the standardized Gaussian variable. The learning rule is basically a Gram-Schmidt-like decorrelation method.

ALGORITHM ASSESSMENT

We comparatively run both PCA and ICA neural algorithms using synthetically generated time series additively corrupted with some white noise to alleviate strict determinism (Table 1 and Fig. 4.). Neural PCA was implemented using the bigradient algorithm since it works for both minimization and maximization of the criterion J_1 under the normality constraints enforced by the penalty function J_2.

The neural ICA algorithms were the extended infomax of Bell and Sejnowski, a semi-adaptive fixed-point fast ICA algorithm (Hyvärinen & Oja, 1997), an adapted variant of EASI algorithm optimized for real data, and the extended generalized lambda distribution (EGLD) maximum likelihood-based algorithm.

In the case of artificially generated sources, the accuracy of separating the latent sources by an algorithm

A

Table 1. The analytical form of the signals sources

Signal sources

Modulated sinusoid: $S(1) = 2 * \sin(t/149) * \cos(t/8)$

Square waves:

$$S(2) = sign\left(\sin\left(12*t + 9*\cos(2/29)\right)\right)$$

Saw-tooth:

$$S(3) = \left(rem(t,79) - 17\right)/23$$

Impulsive curve:

$$S(4) = \left(\left(rem(t,23) - 11\right)/9\right)^5$$

Exponential decay: $S(5) = 5 * \exp(-t/121) * \cos(37*t)$

Spiky noise:

$$S(6) = \left(\left(rand(1,T) < .5\right) * 2 - 1\right) * \log\left(rand(1,T)\right)$$

Figure 4. Sub-Gaussian (left) and super-Gaussian (right) source signals and their corresponding histograms (bottom)
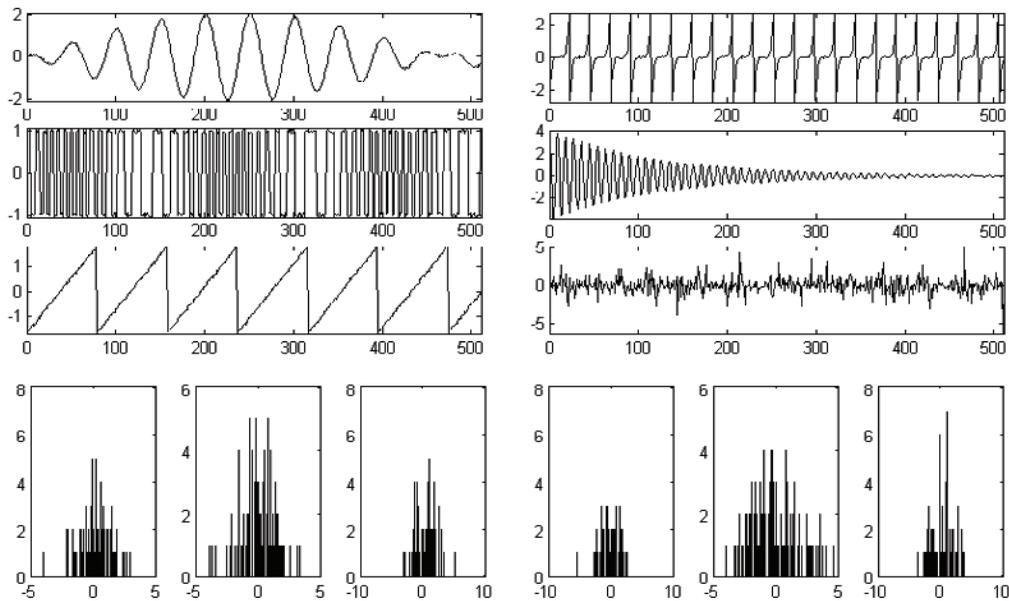
performing ICA can be measured by means of some quantitative indexes. The first we used was defined as the signal-to-interference ratio (*SIR*):

$$SIR = \frac{1}{N} \sum_{i=1}^{N} 10 \cdot \log_{10} \frac{\max\left(Q_i\right)^2}{Q_i^T Q_i - \max\left(Q_i\right)^2}$$

$$(11)$$

where $\mathbf{Q} = \mathbf{BA}$ is the overall transforming matrix of the latent source components, Q_i is the *i*-th column of $\mathbf{Q}$, $\max\left(Q_i\right)$ is the maximum element of Q_i, and *N* is the number of the source signals. The higher the *SIR* is, the better the separation performance of the algorithm.

A secondly employed index was the distance between the overall transforming matrix $\mathbf{Q}$ and an ideal permutation matrix, which is interpreted as the cross-talking error (*CTE*):

$$CTE = \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \frac{|Q_{ij}|}{\max |Q_i|} - 1 \right) + \sum_{j=1}^{N} \left(\sum_{i=1}^{N} \frac{|Q_{ij}|}{\max |Q_j|} - 1 \right)$$

$$(12)$$

Above, Q_{ij} is the *ij*-th element of $\mathbf{Q}$, $\max |Q_i|$ is the maximum absolute valued element of the row *i* in $\mathbf{Q}$, and $\max |Q_j|$ is the maximum absolute valued element of the column *j* in $\mathbf{Q}$. A permutation matrix is defined so that on each of its rows and columns, only one of the elements equals to unity while all the other elements are zero. It means that the *CTE* attains its minimum value zero for an exact permutation matrix (i.e., perfect decomposition) and goes positively higher the more $\mathbf{Q}$ deviates from a permutation matrix (i.e., decomposition of lower accuracy).

We defined the relative signal retrieval error (*SRE*) as the Euclidian distance between the source signals and their best matching estimated components normalized to the number of source

signals, times the number of time samples, and times the module of the source signals:

$$SRE = \frac{1}{TN} \frac{\sqrt{\sum_{i=1}^{N} \left(\sum_{t=1}^{T} \left[x_i(t) - y_i(t) \right]^2 \right)}}{\sqrt{\sum_{i=1}^{N} \left(\sum_{t=1}^{T} \left[x_i(t) \right]^2 \right)}} \ , \ t = 1, 2, ..., T$$

$$(13)$$

The lower the SRE is, the better the estimates approximate the latent source signals.

The stabilized version of FastICA algorithm is attractive by its fast and reliable convergence, and by the lack of parameters to be tuned. The natural gradient incorporated in the BS extended infomax performs better than the original gradient ascent and is computationally less demanding. Though the BS algorithm is theoretically optimal in the sense of dealing with mutual information as objective function, like all neural unsupervised algorithms, its performance heavily depends on the learning rates and its convergence is rather slow. The EGLD algorithm separates skewed distributions, even for zero kurtosis. In terms of computational time, the BS extended infomax algorithm was the fastest, FastICA more faithfully retrieved the sources among all algorithms under test, while the EASI algorithm came out with a full transform matrix $\mathbf{Q}$ that is the closest to unity.

FUTURE TRENDS

Neuromorphic methods in exploratory analysis and data mining are rapidly emerging applications of unsupervised neural training. In recent years, new learning algorithms have been proposed, yet their theoretical properties, range of optimal applicability, and comparative assessment have remained largely unexplored. No convergence theorems are associated with the training algorithms in use. Moreover, algorithm convergence heavily depends on the proper choice of the learning rate(s) and, even when convergence is accomplished, the neural algorithms are relatively slow compared with batch-type computations. Nonlinear and nonstationary neural ICA is expected to be developed due to ANNs

nonalgorithmic processing and their ability to learn nonanalytical relationships if adequately trained.

CONCLUSION

Both PCA and ICA share some common features like aiming at building generative models that are likely to have produced the observed data and performing information preservation and redundancy reduction. In a neuromorphic approach, the model parameters are treated as network weights that are changed during the learning process. The main difficulty in function approximation stems from choosing the network parameters that have to be fixed a priori, and those that must be learnt by means of an adequate training rule.

PCA and ICA have major applications in data mining and exploratory data analysis, such as signal characterization, optimal feature extraction, and data compression, as well as the basis of subspace classifiers in pattern recognition. ICA is much better suited than PCA to perform BSS, blind deconvolution, and equalization.

REFERENCES

Amari, S., Cichocki, A., & Yang, H. (1996). A new learning algorithm for blind source aeparation. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing*, *8*. Cambridge, MA: MIT Press.

Barlow, H. B. (1991). Possible principles underlying the transformation of sensory messages. In W. A. Rosenblith (Ed.), *Sensory communication* (pp. 217-234). Cambridge, MA: MIT Press.

Bell, A., & Sejnowski, T. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, *7*, 1129-1159.

Cardoso, J.-F., & Laheld, B. H. (1996). Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, *44*, 3017-3030.

Cardoso, J.-F. (1998). Blind signal separation: Statistical principles. *Proceeding IEEE*, *9*, 2009-2025.

Comon, P. (1994). Independent component analysis, A new concept? *Signal Processing*, *36*, 287-314.

Esposito, A., Marinaro, M., & Scarpetta, S. (2000). Approximation of continuous and discontinuous mappings by a growing neural RBF-based algorithm. *Neural Networks*, *13*(6) 651-665.

Fiori, S., & Piazza, F. (2000). A general class of APEX-like PCA neural algorithms, *IEEE Transactions on Circuits and Systems - Part I. 47*, 1394-1398.

Gold, S., Rangarajan, A., & Mjolsness, E. (1996). Learning with preknowledge: Clustering with point and graph matching distance. *Neural Computation*, *8*, 787-804.

Haykin, S. (1999). *Neural networks* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Hotelling, H. (1993). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*, 417-441 and 498-520.

Hyvärinen, A., & Oja, E. (1997). A fast fixed-point algorithm for ICA. *Neural Computation*, *9*, 1483-1492.

Karhunen, J. (1996). Neural approaches to independent component analysis and source separation. *Proceedings ESANN'96*, Bruges, Belgium, 249-266.

Lee, T.-W., Girolami, M., Bell, A. J., & Sejnowski, T. J. (2000). A unifying information-theoretic framework for ICA. *Computers and Mathematics with Applications*, *39*, 1-21.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115-133.

Oja, E., Karhunen, J., Wang, L., & Vigario, R. (1995). Principal and independent components in neural networks - Recent developments. *Proceedings VIIth Workshop on Neural Nets*, Vietri, Italy.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*, 267-273.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*, 607-609.

Wang, L. & Karhunen, J. (1996). A unified neural bigradient algorithm for robust PCA and MCA. *International Journal of Neural Sysems, 7*, 53-67.

A

Yang, B. (1995). Projection approximation subspace tracking. *IEEE Transactions on Signal Process*sing, *43*, 1247-1252.

KEY TERMS

Artificial Neural Networks (ANNs): An information-processing synthetic system made up of several simple nonlinear processing units connected by elements that have information storage and programming functions adapting and learning from patterns, which mimics a *biological* neural network.

Blind Source Separation (BSS): Separation of latent nonredundant (e.g., mutually statistically independent or decorrelated) source signals from a set of linear mixtures, such that the regularity of each resulting signal is maximized, and the regularity between the signals is minimized (i.e. statistical independence is maximized) without (almost) any information on the sources.

Confirmatory Data Analysis (CDA): An approach which, subsequent to data acquisition, proceeds with the imposition of a prior model and analysis, estimation, and testing model parameters.

Exploratory Data Analysis (EDA): An approach based on allowing the data itself to reveal its underlying structure and model heavily using the collection of techniques known as *statistical graphics*.

Independent Component Analysis (ICA): An exploratory method for separating a linear mixture of latent signal sources into independent components as optimal estimates of the original sources on the basis of their mutual statistical independence and non-Gaussianity.

Learning Rule: Weight change strategy in a connectionist system aiming to optimize a certain objective function. Learning rules are iteratively applied to the training set inputs with error gradually reduced as the weights are adapting.

Principal Component Analysis (PCA): An orthogonal linear transform based on singular value decomposition that projects data to a subspace that preserves maximum variance.

Adaptive Neuro–Fuzzy Systems

Larbi Esmahi
Athabasca University, Canada

Kristian Williamson
Statistics Canada, Canada

Elarbi Badidi
United Arab Emirates University, UAE

INTRODUCTION

Fuzzy logic became the core of a different approach to computing. Whereas traditional approaches to computing were precise, or hard edged, fuzzy logic allowed for the possibility of a less precise or softer approach (Klir et al., 1995, pp. 212-242). An approach where precision is not paramount is not only closer to the way humans thought, but may be in fact easier to create as well (Jin, 2000). Thus was born the field of soft computing (Zadeh, 1994). Other techniques were added to this field, such as Artificial Neural Networks (ANN), and genetic algorithms, both modeled on biological systems. Soon it was realized that these tools could be combined, and by mixing them together, they could cover their respective weaknesses while at the same time generate something that is greater than its parts, or in short, creating synergy.

Adaptive Neuro-fuzzy is perhaps the most prominent of these admixtures of soft computing technologies (Mitra et al., 2000). The technique was first created when artificial neural networks were modified to work with fuzzy logic, hence the Neuro-fuzzy name (Jang et al., 1997, pp. 1-7). This combination provides fuzzy systems with adaptability and the ability to learn. It was later shown that adaptive fuzzy systems could be created with other soft computing techniques, such as genetic algorithms (Yen et al., 1998, pp. 469-490), Rough sets (Pal et al., 2003; Jensen et al., 2004, Ang et al., 2005) and Bayesian networks (Muller et al., 1995), but the Neuro-fuzzy name was widely used, so it stayed. In this chapter we are using the most widely used terminology in the field.

Neuro-fuzzy is a blanket description of a wide variety of tools and techniques used to combine any aspect of fuzzy logic with any aspect of artificial neural networks. For the most part, these combinations are just extensions of one technology or the other. For example, neural networks usually take binary inputs, but use weights that vary in value from 0 to 1. Adding fuzzy sets to ANN to convert a range of input values into values that can be used as weights is considered a Neuro-fuzzy solution. This chapter will pay particular interest to the sub-field where the fuzzy logic rules are modified by the adaptive aspect of the system.

The next part of this chapter will be organized as follows: in section 1 we examine models and techniques used to combine fuzzy logic and neural networks together to create Neuro-fuzzy systems. Section 2 provides an overview of the main steps involved in the development of adaptive Neuro-fuzzy systems. Section 3 concludes this chapter with some recommendations and future developments.

NEURO-FUZZY TECHNOLOGY

Neuro-fuzzy Technology is a broad term used to describe a field of techniques and methods used to combine fuzzy logic and neural networks together (Jin, 2003, pp. 111-140). Fuzzy logic and neural networks each have their own sets of strengths and weaknesses, and most attempts to combine these two technologies have the goal of using each techniques strengths to cover the others weaknesses.

Neural networks are capable of self-learning, classification and associating inputs with outputs. Neural networks can also become a universal function approximator (Kosko, 1997, pp. 299; Nauck et al., 1998, Nauck et al. 1999). Given enough information about an unknown continuous function, such as its inputs

and outputs, the neural network can be trained to approximate it. The disadvantages of neural networks are they are not guaranteed to converge, that is to be trained properly, and after they have been trained they cannot give any information about why they take a particular course of action when given a particular input.

Fuzzy logic Inference systems can give human readable and understandable information about why a particular course of action was taken because it is governed by a series of IF THEN rules. Fuzzy logic systems can adapt in a way that their rules and the parameters of the fuzzy sets associated with those rules can be changed to meet some criteria. However fuzzy logic systems lack the capability for self-learning, and must be modified by an external entity. Another salient feature of fuzzy logic systems is that they are, like artificial neural networks, capable of acting as universal approximators.

The common feature of being able to act as a universal approximator is the basis of most attempts to merge these two technologies. Not only it can be used to approximate a function but it can also be used by both neural networks, and fuzzy logic systems to approximate each other as well. (Pal et al., 1999, pp. 66)

Universal approximation is the ability of a system to replicate a function to some degree. Both neural networks and fuzzy logic systems do this by using a non-mathematical model of the system (Jang et al., 1997, pp. 238; Pal et al., 1999, pp. 19). The term approximate is used as the model does not have to match the simulated function exactly, although it is sometime possible to do so if enough information about the function is available. In most cases it is not necessary or even desirable to perfectly simulate a function as this takes time and resources that may not be available and close is often good enough.

Categories of Neuro-Fuzzy Systems

Efforts to combine fuzzy logic and neural networks have been underway for several years and many methods have been attempted and implemented. These methods are of two major categories:

- Fuzzy Neural Networks (FNN): are neural networks that can use fuzzy data, such as fuzzy rules, sets and values (Jin, 2003, pp.205-220).

- Neural-Fuzzy Systems (NFS): are fuzzy systems "augmented" by neural networks (Jin, 2003, pp.111-140).

There also four main architectures used for implementing neuro-fuzzy systems:

- Fuzzy Multi-layer networks (Jang, 1993; Mitra et al., 1995; Mitra et al., 2000; Mamdani et al., 1999; Sugeno et al., 1988, Takagi et al., 1985).
- Fuzzy Self-Organizing Map networks (Drobics et al., 2000; Kosko, 1997, pp. 98; Haykin, 1999, pp. 443)
- Black-Box Fuzzy ANN (Bellazzi et al., 1999; Qiu, 2000; Monti, 1996)
- Hybrid Architectures (Zatwarnicki, 2005; Borzemski et al., 2003; Marichal et al., 2001; Rahmoun et al., 2001; Koprinska et al., 2000; Wang et al. 1999; Whitfort et al., 1995).

DEVELOPMENT OF ADAPTIVE NEURO-FUZZY SYSTEMS

Developing an Adaptive Neuro-fuzzy system is a process that is similar to the procedures used to create fuzzy logic systems, and neural networks. One advantage of this combined approach is that it is usually no more complicated than either approach taken individually.

As noted above, there are two methods of creating a Neuro-fuzzy system; integrating fuzzy logic into a neural network framework (FNN), and implementing neural networks into a fuzzy logic system (NFS). A fuzzy neural network is just a neural network with some fuzzy logic components; hence is generally trained like a normal neural network is.

Training Process: The training regimen for a NFS differs slightly from that used to create a neural network and a fuzzy logic system in some key ways, while at the same time incorporating many improvements over those training methods.

The training process of a Neuro-fuzzy system has five main steps: (Von Altrock, 1995, pp. 71-75)

- **Obtain Training Data:** The data must cover all possible inputs and output, and all the critical regions of the function if it is to model it in an appropriate manner.

- **Create a Fuzzy Logic System:** The fuzzy system may be an existing system which is known to work, such as one that has been in production for some time or one that has been created by following expert system development methodologies.
- **Define the Neural Fuzzy Learning:** This phase deals with defining what you want the system to learn. This allows greater control over the learning process while still allowing for rule knowledge discovery.
- **Training Phase:** To run the training algorithm. The algorithm may have parameters that can be adjusted to modify how the system is to be modified during training.
- **Optimization and Verification:** Validation can take many forms, but will usually involve feeding the system a series of known inputs to determine if the system generates the desired output, and or is within acceptable parameters. Furthermore, the rules and membership functions may be extracted so they can be examined by human experts for correctness.

CONCLUSION AND FUTURE DEVELOPMENTS

Advantages of ANF systems: Although there are many ways to implement a Neuro-fuzzy system, the advantages described for these systems are remarkably uniform across the literature. The advantages attributed to Neuro-fuzzy systems as compared to ANNs are usually related to the following aspects:

- **Faster to train:** This is due to the massive number of connections present in the ANN, and the non-trivial number of calculations associated with each. As well, most neural fuzzy systems can be trained by going through the data once, whereas a neural network may need to be exposed to the same training data many times before it converges.
- **Less computational resources:** Neural fuzzy system is smaller in size and contains fewer internal connections than a comparable ANN, hence it is faster and use significantly less resources.
- **Offer the possibility to extract the rules:** This is a major advantage over ANNs in that the rules governing a system can be communicated to the human users in an easily understandable form.

Limitation of ANF systems: The greatest limitation in creating adaptive systems is known as the "Curse of Dimensionality", which is named after the exponential growth in the number of features that the model has to keep track of as the number of input attributes increases. Each attribute in the model is a variable in the system, which corresponds to an axis in a multidimensional graph that the function is mapped into. The connections between different attributes correspond to the number of potential rules in the system as given by the formula:

$$N_{rules} = (L_{lingustic\_terms})^{variables} \text{ (Gorrostieta et al., 2006)}$$

This formula becomes more complicated if there are different numbers of linguistic variables (fuzzy sets) covering each attribute dimension. Fortunately there are ways around this problem. As the neural fuzzy system is only approximating the function being modeled, the system may not need all the attributes to achieve the desired results.

Another area of criticism in the Neuro-fuzzy field is related to aspects that can't be learned or approximated. One of the most known aspects here is the caveat attached to the universal approximation. In fact, the function being approximated has to be continuous; a continuous function is a function that does not have a singularity, a point where it goes to infinity. Other functions that Adaptive Neuro-fuzzy systems may have problems learning are things like encryption algorithms, which are purposely designed to be resistant to this type of analysis.

Future developments: Predicting the future has always been hard; however for ANF technology the future expansion has been made easy because of the widespread use of its basis technology (neural networks and fuzzy logic). Mixing of these technologies creates synergies as they remediate to each other weaknesses. ANF technology allows complex system to be grown instead of someone having to build them.

One of the most promising areas for ANF systems is System Mining. There exist many cases where we wish to automate a system that cannot be systematically described in a mathematical manner. This means there is no way of creating a system using classical development methodologies (i.e. Programming a simulation.). If we have an adequately large set of examples of inputs and their corresponding outputs, ANF can be used to get a model of the system. The rules and their associated

fuzzy sets can then be extracted from this system and examined for details about how the system works. This knowledge can be used to build the system directly. One interesting application of this technology is to audit existing complex systems. The extracted rules could be used to determine if the rules match the exceptions of what the system is supposed to do, and even detect fraud actions. Alternatively, the extracted model may show an alternative, and or more efficient manner of implementing the system.

REFERENCES

Ang, K. K. & Quek, C. (2005). RSPOP: Rough Set-Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. Neural Computation, (17) 1, 205-243.

Bellazzi, R., Guglielmann, R. & Ironi L. (1999). A qualitative-fuzzy framework for nonlinear black-box system identification. In Dean T., editor, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI 99), volume 2, pages 1041—1046. Morgan Kaufmann Publishers.

Borzemski, L. & Zatwarnicki, K. (2003). A fuzzy adaptive request distribution algorithm for cluster-based Web systems. In the Proceedings Eleventh Euromicro Conference on Parallel, Distributed and Network-Based Processing, 119 - 126. Institute of Electrical & Electronics Engineering Publisher.

Chavan, S., Shah, K., Dave, N., Mukherjee, S., Abraham, A., & Sanyal, S. (2004). Adaptive neuro-fuzzy intrusion detection systems. In Proceedings of the International Conference on Information Technology: Coding and Computing, ITCC 2004, 70 - 74 Vol.1. Institute of Electrical & Electronics Engineering Publisher.

Drobics, M., Winiwater & W., Bodenhofer, U. (2000). Interpretation of self-organizing maps with fuzzy rules. In Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'00), p. 0304. IEEE Computer Society Press.

Gorrostieta, E. & Pedraza, C. (2006). Neuro Fuzzy Modeling of Control Systems. In Proceedings of the 16th IEEE International Conference on Electronics, Communications and Computers (CONIELECOMP 2006), 23 – 23. IEEE Computer Society Publisher.

Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. Prentice Hall Publishers, 2nd edition

Jang, J. S. R., Sun C. T. & Mizutani E. (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall Publishers, US Ed edition.

Jang, J.-S.R. (1993). ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics, (23) 3, 665 – 685.

Jensen, R. & Shen, Q. (2004). Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Transactions on Knowledge and Data Engineering, (16) 12, 1457 – 1471.

Jin Y. (2000). Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement. IEEE Transactions on Fuzzy Systems, (8) 2, 212-221.

Jin, Y. (2003). Advanced Fuzzy Systems Design and Applications. Physica-Verlag Heidelberg Publishers; 1 edition.

Klir, G. J. & Yuan, B. (1995). Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall PTR Publishers; 1st edition.

Koprinska, L. & Kasabov, N. (2000). Evolving fuzzy neural network for camera operations recognition. In Proceedings of the 15th International Conference on Pattern Recognition, 523 - 526 vol.2. IEEE Computer Society Press Publisher.

Kosko, B. (1997). Fuzzy Engineering. Prentice Hall Publishers, 1st edition.

Mamdani, E. H. & Assilian, S. (1999). An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. International Journal of Human-Computer Studies, (51) 2, 135-147.

Marichal, G.N., Acosta, L., Moreno, L., Mendez, J.A. & Rodrigo, J. J. (2001). Obstacle Avoidance for a Mobile Robot: A neuro-fuzzy approach. Fuzzy Sets and Systems, (124) 2, 171- 179.

Mitra, S. & Hayashi Y. (2000). Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks, (11) 3, 748 – 768.

Mitra, S. & Pal, S. K. (1995). Fuzzy multi-layer perceptron, inferencing and rule generation. IEEE Transactions on Neural Networks, (6) 1, 51-63.

Monti, A. (1996). A fuzzy-based black-box approach to IGBT modeling. In Proceedings of the Third IEEE International Conference on Electronics, Circuits, and Systems: ICECS '96. 1147 - 1150 vol.2. Institute of Electrical & Electronics Engineering Publisher.

Muller, P. & Insua, D.R. (1998). Issues in Bayesian Analysis of Neural Network Models. Neural Computation (10) 3, 749-770.

Nauck, D. & Kruse R. (1999). Neuro-fuzzy systems for function approximation. Fuzzy Sets and Systems (101) 261-271.

Nauck, D. & Kruse, R. (1998). A neuro-fuzzy approach to obtain interpretable fuzzy systems for function approximation. In Wcci 98: Proceedings of Fuzz-IEEE '98, 1106 - 1111 vol.2. IEEE World Congress on Computational Intelligence. Institute of Electrical & Electronics Engineering Publisher.

Pal, S. K. & Mitra S. (1999). Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing. John Wiley & Sons Publishers, 1st edition.

Pal, S.K., Mitra, S. & Mitra, P. (2003). Rough-fuzzy MLP: modular evolution, rule generation, and evaluation. IEEE Transactions on Knowledge and Data Engineering, (15) 1, 14 – 25.

Qiu F. (2000). Opening the black box of neural networks with fuzzy set theory to facilitate the understanding of remote sensing image processing. In Proceedings of the IEEE 2000 International Geoscience and Remote Sensing Symposium: Taking the Pulse of the Planet: The Role of Remote Sensing in Managing the Environment, IGARSS 2000. 1531 - 1533 vol.4. Institute of Electrical & Electronics Engineering Publisher.

Rahmoun, A. & Berrani, S. (2001). A genetic-based neuro-fuzzy generator: NEFGEN. ACS/IEEE International Conference on Computer Systems and Applications, 18 – 23. Institute of Electrical & Electronics Engineering Publisher.

Sugeno, M. & Kang, G. T. (1998). Structure identification of fuzzy model. Fuzzy Sets and Systems, (28) 1, 15-33.

Takagi T. & Sugeno M. (1985). Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics, (15), 116-132.

Von Altrock, C. (1995). Fuzzy Logic and Neuro Fuzzy Applications Explained. Prentice Hall Publishers.

Wang L. & Yen J. (1999). Extracting Fuzzy Rules for System Modeling Using a Hybrid of Genetic Algorithms and Kalman Filter. Fuzzy Sets Systems, (101) 353–362.

Whitfort, T., Matthews, C. & Jagielska, I. (1995). Automated knowledge acquisition for a fuzzy classification problem. In Kasabov, N. K. & Coghill, G. (Editors), Proceedings of the Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, 227 – 230. IEEE Computer Society Press Publisher.

Yen, J. & Langari, R. (1998). Fuzzy Logic: Intelligence, Control, and Information. Prentice Hall Publishers.

Zadeh, L. A. (1994). Fuzzy Logic, Neural Networks, and Soft Computing. Communications of the ACM (37) 3, 77-84.

Zatwarnicki, K. (2005). Proposal of a neuro-fuzzy model of a WWW server. Proceedings of the Fifth International Conference on Intelligent Systems Design and Applications ISDA '05, 141 – 146. Institute of Electrical & Electronics Engineering Publisher.

KEY TERMS

Artificial Neural Networks (ANN): An artificial neural network, often just called a "neural network" (NN), is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing based on a connectionist approach to computation. Knowledge is acquired by the network from its environment through a learning process, and interneuron connection strengths (synaptic weighs) are used to store the acquired knowledge.

Evolving Fuzzy Neural Network (EFuNN): An Evolving Fuzzy Neural Network is a dynamic architecture where the rule nodes grow if needed and shrink by aggregation. New rule units and connections can be added easily without disrupting existing nodes.

The learning scheme is often based on the concept of "winning rule node".

Fuzzy Logic: Fuzzy logic is an application area of fuzzy set theory dealing with uncertainty in reasoning. It utilizes concepts, principles, and methods developed within fuzzy set theory for formulating various forms of sound approximate reasoning. Fuzzy logic allows for set membership values to range (inclusively) between 0 and 1, and in its linguistic form, imprecise concepts like "slightly", "quite" and "very". Specifically, it allows partial membership in a set.

Fuzzy Neural Networks (FNN): are Neural Networks that are enhanced with fuzzy logic capability such as using fuzzy data, fuzzy rules, sets and values.

Neuro-Fuzzy Systems (NFS): A neuro-fuzzy system is a fuzzy system that uses a learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples.

Self-Organizing Map (SOM): The self-organizing map is a subtype of artificial neural networks. It is trained using unsupervised learning to produce low dimensional representation of the training samples while preserving the topological properties of the input space. The self-organizing map is a single layer feed-forward network where the output syntaxes are arranged in low dimensional (usually 2D or 3D) grid. Each input is connected to all output neurons. Attached to every neuron there is a weight vector with the same dimensionality as the input vectors. The number of input dimensions is usually a lot higher than the output grid dimension. SOMs are mainly used for dimensionality reduction rather than expansion.

Soft Computing: Soft Computing refers to a partnership of computational techniques in computer science, artificial intelligence, machine learning and some engineering disciplines, which attempt to study, model, and analyze complex phenomena. The principle partners at this juncture are fuzzy logic, neuron-computing, probabilistic reasoning, and genetic algorithms. Thus the principle of soft computing is to exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low cost solution, and better rapport with reality.

Adaptive Technology and Its Applications

João José Neto
Universidade de São Paulo, Brazil

INTRODUCTION

Before the advent of software engineering, the lack of memory space in computers and the absence of established programming methodologies led early programmers to use self-modification as a regular coding strategy.

Although unavoidable and valuable for that class of software, solutions using self-modification proved inadequate while programs grew in size and complexity, and security and reliability became major requirements.

Software engineering, in the 70's, almost led to the vanishing of self-modifying software, whose occurrence was afterwards limited to small low-level machine-language programs with very special requirements.

Nevertheless, recent research developed in this area, and the modern needs for powerful and effective ways to represent and handle complex phenomena in high-technology computers are leading self-modification to be considered again as an implementation choice in several situations.

Artificial intelligence strongly contributed for this scenario by developing and applying non-conventional approaches, e.g. heuristics, knowledge representation and handling, inference methods, evolving software/hardware, genetic algorithms, neural networks, fuzzy systems, expert systems, machine learning, etc.

In this publication, another alternative is proposed for developing Artificial Intelligence applications: the use of adaptive devices, a special class of abstractions whose practical application in the solution of current problems is called Adaptive Technology.

The behavior of adaptive devices is defined by a dynamic set of rules. In this case, knowledge may be represented, stored and handled within that set of rules by adding and removing rules that represent the addition or elimination of the information they represent.

Because of the explicit way adopted for representing and acquiring knowledge, adaptivity provides a very simple abstraction for the implementation of artificial learning mechanisms: knowledge may be comfortably gathered by inserting and removing rules, and handled by tracking the evolution of the set of rules and by interpreting the collected information as the representation of the knowledge encoded in the rule set.

MAIN FOCUS OF THIS ARTICLE

This article provides concepts and foundations on adaptivity and adaptive technology, gives a general formulation for adaptive abstractions in use and indicates their main applications.

It shows how rule-driven devices may turn into adaptive devices to be applied in learning systems modeling, and introduces a recently formulated kind of adaptive abstractions having adaptive subjacent devices. This novel feature may be valuable for implementing meta-learning, since it enables adaptive devices to change dynamically the way they modify their own set of defining rules.

A significant amount of information concerning adaptivity and related subjects may be found at the (LTA Web site).

BACKGROUND

This section summarizes the foundations of adaptivity and establishes a general formulation for adaptive rule-driven devices (Neto, 2001), non-adaptivity being the only restriction imposed to the subjacent device.

Some theoretical background is desirable for the study and research on adaptivity and Adaptive Technology: formal languages, grammars, automata, computation models, rule-driven abstractions and related subjects.

Nevertheless, either for programming purposes or for an initial contact with the theme, it may be unproblematic to catch the basics of adaptivity even having no prior expertise with computer-theoretical subjects.

In adaptive abstractions, adaptivity may be achieved by attaching adaptive actions to selected rules chosen

from the rule set defining some subjacent non-adaptive device.

Adaptive actions enable adaptive devices to dynamically change their behavior without external help, by modifying their own set of defining rules whenever their subjacent rule is executed.

For practical reasons, up to two adaptive actions are allowed: one to be performed prior to the execution of its underlying rule, and the other, after it.

An adaptive device behaves just as it were piecewise non-adaptive: starting with the configuration of its initial underlying device, it iterates the following two steps, until reaching some well-defined final configuration:

- While no adaptive action is executed, run the underlying device;
- Modify the set of rules defining the device by executing an adaptive action.

Rule-Driven Devices

A *rule-driven device* is any formal abstraction whose behavior is described by a rule set that maps each possible configuration of the device into a corresponding next one.

A device is *deterministic* when, for any configuration and any input, a single next configuration is possible. Otherwise, it is said *non-deterministic*.

Non-deterministic devices allow multiple valid possibilities for each move, and require backtracking, so deterministic equivalents are usually preferable in practice.

Assume that:

- D is some rule-driven device, defined as $D = (C, R, S, c_0, A)$.
- C is its set of possible configurations.
- $R \subseteq C \times (S \cup \{\varepsilon\}) \times C$ is the set of rules describing its behavior, where ε denotes empty stimulus, representing no events at all.
- S is its set of valid input stimuli.
- $c_0 \in C$ is its initial configuration.
- $A \subseteq C$ is its set of final configurations.

Let $c_i \Rightarrow^{(r)} c_{i+1}$ (for short, $c_i \Rightarrow c_{i+1}$) denote the application of some rule $r = (c_i, s, c_{i+1}) \in R$ to the current

configuration c_i in response to some input stimulus $s \in S \cup \{\varepsilon\}$, yielding its next configuration c_{i+1}.

Successive applications of rules in response to a stream $w \in S*$ of input stimuli, starting from the initial configuration c_0 and leading to some final configuration $c \in A$ is denoted $c_0 \Rightarrow^*_w c$ (The star postfix operator in the formulae denotes the Kleene closure: its preceding element may be re-instantiated or reapplied an arbitrary number of times).

We say that D defines a sentence w if, and only if, $c_0 \Rightarrow^*_w c$ holds for some $c \in A$. The collection $L(D)$ of all such sentences is called the language defined by D:

$$L(D) = \{w \in S* \mid c_0 \Rightarrow^*_w c, c \in A\}.$$

Adaptive (Rule-Driven) Devices

An adaptive rule-driven device $AD = (ND_0, AM)$ associates an initial subjacent rule-driven device $ND_0 = (C, NR_0, S, c_0, A)$, to some adaptive mechanism AM, that can dynamically change its behavior by modifying its defining rules.

That is accomplished by executing non-null adaptive actions chosen from a set AA of *adaptive actions*, which includes the *null adaptive action* a^0.

A built-in counter t starts at 0 and is self-incremented upon any adaptive actions' execution. Let X_j denote the value of X after j executions of adaptive actions by AD.

Adaptive actions in AA call functions that map AD current set AR_t of adaptive rules into AR_{t+1} by inserting to and removing adaptive rules ar from AM.

Let **AR** be the set of all possible sets of adaptive rules for AD. Any $a^k \in A$ maps the current set of rules $AR_t \in \mathbf{AR}$ into $AR_{t+1} \in \mathbf{AR}$:

$$a^k : \mathbf{AR} \to \mathbf{AR}$$

AM associates to each rule $nr^p \in NR$ of AD underlying device ND a pair of adaptive actions $ba^p, aa^p \in AA$:

$$AM \subseteq AA \times NR \times AA$$

Notation

When writing elementary adaptive actions, $?[ar]$, $+[ar]$ and $-[ar]$ respectively denote searching, inserting and eliminating adaptive rules that follow template *ar*.

Note that *ar* may contain references to parameters, variables and generators, in order to allow cross-referencing among elementary adaptive actions inside an adaptive function.

Given an underlying rule $nr^p \in NR$, we define an adaptive rule $ar^p \in AM$ as:

$$ar^p = \left(ba^p, nr^p, aa^p \right)$$

For each *AD* move, *AM* applies some ar^p in three steps:

a. execution of adaptive action ba^p before applying the subjacent rule nr^p;
b. application of the underlying non-adaptive rule nr^p;
c. execution of adaptive action aa^p.

The following algorithm sketches the overall operation of *AD*:

1. Initialize c_0, *w*;
2. If *w* is exhausted, go to 7 else get next event s_t;
3. For the current configuration c_t, determine the set *CR* of c_t-compatible rules;

a. if $CR = \varnothing$, reject *w*.
b. if $CR = \{(c_t, s, c')\}$, apply (c_t, s, c') as in steps 4-6, leading *AD* to $c_{t+1} = c'$.
c. if $CR = \{r^k = (c_t, s, c^k) \mid c^k \in C, k = 1, \dots, n,\ n > 1\}$, apply all rules r^k in parallel, as in steps 4-6, leading *AD* to $c^1, c^2, \dots, c^n$, respectively.

4. If $ba^p = a^0$, go to 2, else apply first ba^p. If rule ar^p were removed by ba^p, go to 3 aborting ar^p, else *AD* reached an intermediate configuration, then go to 2.
5. Apply nr^p to the current (intermediate) configuration, yielding a new intermediate configuration;

6. Apply aa^p, yielding the next (stable) configuration for *AD*; go to 2
7. If some $c_{t+1} \in F$ was reached, then *AD* accepts *w*, otherwise *AD* rejects *w*; stop.

Hierarchical Multi-Level Adaptive Devices

Let us define a more elaborated adaptive device by generalizing the definition above. Call non-adaptive devices *level-0 devices*; define *level-1 devices* those having subjacent level-0 devices, to each of whose rules a pair of level-1 adaptive actions are attached.

Let the subjacent device be some level-k adaptive device. One may construct a level-(*k*+1) device attaching a pair of level-(*k*+1) adaptive actions to each of its rules. This is the induction step for the definition of hierarchically structured multi-level adaptive devices.

Besides the set of rules defining the subjacent level-*k* device, for $k > 0$, adaptive functions' subjacent device performs at its own level, which may use level-(*k*+1) adaptive actions to modify the behavior of level-*k* adaptive functions.

So, for $k > 0$, level-(*k*+1) devices can change the way their subjacent level-*k* devices modify themselves. That also holds for $k = 1$, since even for $k = 0$ the (empty) set of adaptive functions still exists.

Notation

The absence of adaptive actions in non-adaptive rules nr is explicitly expressed by stating all level-0 rules r_0 in the form $\left(a^0\ nr\ a^0 \right)$. Therefore, level-*k* rules r_k take the general format $\left(b_k\ r_{k-1}\ a_k \right)$, with both b_k and a_k level-*k* adaptive actions for any adaptive level $k \geq 0$.

So, level-*k* adaptive devices have all their defining rules stated in the standard form

$$\left(b_k \left(b_{k-1} \left(\dots \left(b_1 \left(a^0 (c, \sigma, c') a^0 \right) a_1 \right) \dots \right) a_{k-1} \right) a_k \right),$$

with

$$\left(b_{k-1} \left(\dots \left(b_1 \left(a^0 (c, \sigma, c') a^0 \right) a_1 \right) \dots \right) a_{k-1} \right)$$

representing one of the rules defining the subjacent level-(*k* – 1) adaptive device.

A

Hence, level-*i* adaptive actions can modify both the set of level-*i* adaptive rules and the set of elementary adaptive actions defining level-$(i-1)$ adaptive functions.

A SIMPLE ILLUSTRATIVE EXAMPLE

In the following example, graphical notation is used for clarity and conciseness. When drawing automata, (as usual) circles represent states; double-line circles indicate final states; arrows indicate transitions; labels on the arrows indicate tokens consumed by the transition and (optionally) an associated adaptive action. When representing adaptive functions, automata fragments in brackets stand for a group of transitions to be added (+) or removed (-) when the adaptive action is applied.

Figure 1 shows the starting shape of an adaptive automaton that accepts $a^n b^{2n} c^{3n}$, n≥0. At state **1**, it includes a transition consuming **a**, which performs adaptive action $\mathcal{A}(\,)$.

Figure 2 defines how $\mathcal{A}(\,)$ operate:

Figure 1. Initial configuration of the illustrative adaptive automaton

- Using state **2** as reference, eliminate empty transitions using states **x** and **y**
- Add a sequence starting at **x**, with two transitions consuming **b**
- Append the sequence of two empty transitions sharing state **2**

Append a sequence with three transitions consuming **c,** ending at **y.**

Figure 3 shows the first two shape changes of this automaton after consuming the two first symbols **a** (at state **1**) in sentence $a^2 b^4 c^6$. In its last shape, the automaton trivially consumes the remaining $b^4 c^6$, and does not change any more.

There are many other examples of adaptive devices in the references. This almost trivial and intuitive case was shown here for illustration purposes only.

Knowledge Representation

The preceding example illustrates how adaptive devices use the set of rules as their only element for representing and handling knowledge.

A rule (here, a transition) may handle parametric information in its components (here, the transition's origin and destination states, the token labeling the transition, the adaptive function it calls, etc.).

Rules may be combined together in order to represent some non-elementary information (here, the sequences of transitions consuming tokens "b" and "c" keep track of the value of n in each particular sentence). This way, rules and their components may work and may be interpreted as low-level elements of knowledge.

Although being impossible to impose rules on how to represent and handle knowledge in systems repre-

Figure 2. Adaptive function $\mathcal{A}(\,)$

Figure 3. Configurations of the adaptive automaton after executing A ( ) once and twice
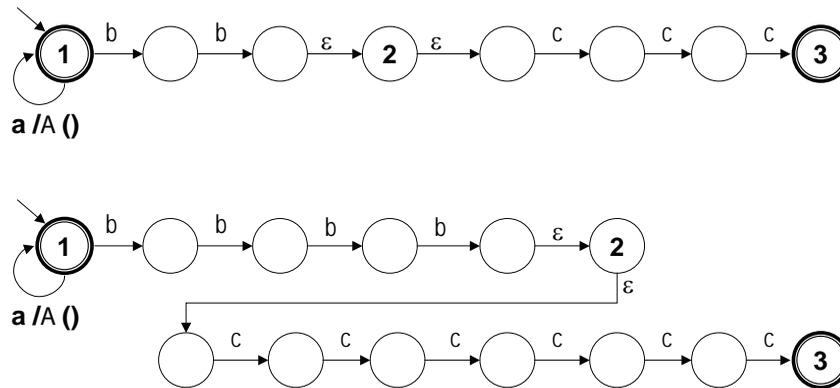
sented with adaptive devices, the details of the learning process may be chosen according to the particular needs of each system being modeled.

In practice, the learning behavior of an adaptive device may be identified and measured by tracking the progress of the set of rules during its operation and interpreting the dynamics of its changes.

In the above example, when transitions are added to the automaton by executing adaptive action $\mathcal{A}(\,)$, one may interpret the length of the sequence of transitions consuming "b" (or "c") as a manifestation of the knowledge that is being gathered by the adaptive automaton on the value of n (its exact value becomes available after the sub-string of tokens "a" is consumed).

FUTURE TRENDS

Adaptive abstractions represent a significant theoretical advance in Computer Science, by introducing and exploring powerful non-classical concepts such as: time-varying behavior, autonomously dynamic rule sets, multi-level hierarchy, static and dynamic adaptive actions.

Those concepts allow establishing a modeling style, proper for describing complex learning systems, for efficiently solving traditionally hard problems, for dealing with self-modifying learning methods, and for providing computer languages and environments for comfortable elaboration of quality programs with dynamically-variant behavior.

All those features are vital for conceiving, modeling, designing and implementing applications in Artificial Intelligence, which benefits from adaptivity while expressing traditionally difficult-to-describe Artificial Intelligence facts.

Listed below are features Adaptive Technology offers to several fields of Computation, especially to Artificial Intelligence-related ones, indicating their main impacts and applications.

- Adaptive Technology provides a true computation model, constructed around formal foundations. Most Artificial Intelligence techniques in use are very hard to express and follow since the connection between elements of the models and information they represent is often implicit, so their operation reasoning is difficult for a human to track and plan. Adaptive rule-driven devices concentrate all stored knowledge in their rules, and the whole logic that handles such information, in their adaptive actions. Such properties open for Artificial Intelligence the possibility to observe, understand and control adaptive-device-modeled phenomena. By following and interpreting how and why changes occur in the device set of rules, and by tracking semantics of adaptive actions, one can infer the reasoning of the model reactions to its input.
- Adaptive devices have enough processing power to model complex computations. In (Neto, 2000) some well-succeeded use cases are shown with

simple and efficient adaptive devices used instead of complex traditional formulations.

- Adaptive Devices are Turing Machine-equivalent computation models that may be used in the construction of single-notation full specifications of programming languages, including lexical, syntactical, context-dependent static-semantic issues, language built-in features such as arithmetic operations, libraries, semantics, code generation and optimization, run-time code interpreting, etc.
- Adaptive devices are well suited for representing complex languages, including idioms. Natural language particularly require several features to be expressed and handled, as word inflexions, orthography, multiple syntax forms, phrase ordering, ellipsis, permutation, ambiguities, anaphora and others. A few simple techniques allow adaptive devices to deal with such elements, strongly simplifying the effort of representing and processing them. Applications are wide, including machine translation, data mining, text-voice and voice-text conversion, etc.
- Computer art is another fascinating potential application of adaptive devices. Music and other artistic expressions are forms of human language. Given some language descriptions, computers can capture human skills and automatically generate interesting outputs. Well-succeeded experiments were carried out in the field of music, with excellent results (Basseto, 1999).
- Decision-taking systems may use Adaptive Decision Tables and Trees for constructing intelligent systems that accept training patterns, learn how to classify them, and therefore, classify unknown patterns. Well-succeeded experiments include: classifying geometric patterns, decoding sign languages, locating patterns in images, generating diagnoses from symptoms and medical data, etc.
- Language inference uses Adaptive Devices to generate formal descriptions of languages from samples, by identifying and collecting structural information and generalizing on the evidence of repetitive or recursive constructs (Matsuno, 2006).
- Adaptive Devices can be used for learning purposes by storing as rules the gathered information on some monitored phenomenon. In educational

systems, the behavior of both students and trainers can be inferred and used to decide how to proceed.
- One can construct Adaptive Devices whose underlying abstraction is a computer language. Statements in such languages may be considered as rules defining behavior of a program. By attaching adaptive rules to statements, the program becomes self-modifiable. Adaptive languages are needed for adaptive applications to be expressed naturally. For adaptivity to become a true programming style, techniques and methods must be developed to construct good adaptive software, since adaptive applications developed so far were usually produced in strict ad-hoc way.

CONCLUSION

Adaptive Technology concerns techniques, methods and subjects referring to actual application of adaptivity.

Adaptive automata (Neto, 1994) were first proposed for practical representation of context-sensitive languages (Rubinstein, 1995). Adaptive grammars (Iwai, 2000) were employed as its generative counterpart (Burshteyn, 1990), (Christiansen, 1990), (Cabasino, 1992), (Shutt, 1993), (Jackson, 2006).

For specification and analysis of real time reactive systems, works were developed based on adaptive versions of statecharts (Almeida Jr., 1995), (Santos, 1997). An interesting confirmation of power and usability of adaptive devices for modeling complex systems (Neto, 2000) was the successful use of Adaptive Markov Chains in a computer music-generating device (Basseto, 1999).

Adaptive Decision Tables (Neto, 2001) and Adaptive Decision Trees (Pistori, 2006) are nowadays being experimented in decision-taking applications.

Experiments have been reported that explore the potential of adaptive devices for constructing language inference systems (Neto, 1998), (Matsuno, 2006).

An important area in which adaptive devices shows its strength is the specification and processing of natural languages (Neto, 2003). Many other results are being achieved while representing syntactical context-dependencies of natural language.

Simulation and modeling of intelligent systems are other concrete applications of adaptive formalisms, as illustrated in the description of the control mechanism

of an intelligent autonomous vehicle which collects information from its environment and builds maps for navigation.

Many other applications for adaptive devices are possible in several fields.

REFERENCES

(* or ** - downloadable from LTA Website; ** - in Portuguese only)

Almeida Jr., J.R. (1995)**. STAD - *Uma ferramenta para representação e simulação de sistemas através de statecharts adaptativos.* São Paulo, 202p. Doctoral Thesis. Escola Politécnica, Universidade de São Paulo.

Basseto, B.A., Neto, J.J. (1999)*. *A stochastic musical composer based on adaptive algorithms.* Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação. SBC-99, Vol. 3, pp. 105-13.

Burshteyn, B. (1990). *Generation and recognition of formal languages by modifiable grammars.* ACM SIGPLAN Notices, v.25, n.12, p.45-53, 1990.

Cabasino, S.; Paolucci, P.S.; Todesco, G.M. (1992). *Dynamic parsers and evolving grammars.* ACM SIGPLAN Notices, v.27, n.11, p.39-48, 1992.

Christiansen, H. (1990). *A survey of adaptable grammars.* ACM SIGPLAN Notices, v.25, n.11, p.33-44.

Iwai, M.K. (2000)**. *Um formalismo gramatical adaptativo para linguagens dependentes de contexto.* São Paulo 2000, 191p. Doctoral Thesis. Escola Politécnica, Universidade de São Paulo.

Jackson, Q.T. (2006). *Adapting to Babel – Adaptivity and context-sensitivity parsing: from $a^nb^nc^n$ to RNA* – A Thotic Technology Partners Research Monograph.

LTA Website: http://www.pcs.usp.br/~lta

Matsuno, I.P. (2006)**. *Um Estudo do Processo de Inferência de Gramáticas Regulares e Livres de Contexto Baseados em Modelos Adaptativos.* M.Sc. Dissertation, Escola Politécnica, Universidade de São Paulo.

Neto, J.J.; Moraes, M.de. (2003)* *Using Adaptive Formalisms to Describe Context-Dependencies in Natural Language.* Computational Processing of the Portuguese Language 6th International Workshop, PROPOR 2003, LNAI Volume 2721, Faro, Portugal, June 26-27, Springer-Verlag, 2003, pp 94-97.

Neto, J. J. (2001)*. *Adaptive Rule-Driven Devices - General Formulation and Case Study.* Lecture Notes in Computer Science. Watson, B.W. and Wood, D. (Eds.): Implementation and Application of Automata - 6th International Conference, CIAA 2001, Vol.2494, Pretoria, South Africa, July 23-25, Springer-Verlag, 2001, pp. 234-250.

Neto, J.J. (1994)*. *Adaptive automata for context-dependent languages.* ACM SIGPLAN Notices, v.29, n.9, p.115-24, 1994.

Neto, J.J. (2000)*. *Solving Complex Problems Efficiently with Adaptive Automata.* CIAA 2000 - Fifth International Conference on Implementation and Application of Automata - London, Ontario, Canada.

Neto, J.J., Iwai, M.K. (1998)*. *Adaptive automata for syntax learning.* XXIV Conferencia Latinoamericana de Informática CLEI'98, Quito - Ecuador, tomo 1, pp.135-146.

Pistori, H.; Neto, J.J.; Pereira, M.C. (2006)* *Adaptive Non-Deterministic Decision Trees: General Formulation and Case Study.* INFOCOMP Journal of Computer Science, Lavras, MG.

Rubinstein, R.S.; Shutt. J.N. (1995). *Self-modifying finite automata: An introduction*, Information processing letters, v.56, n.4, 24, p.185-90.

Santos, J.M.N. (1997)**. *Um formalismo adaptativo com mecanismos de sincronização para aplicações concorrentes.* São Paulo, 98p. M.Sc. Dissertation. Escola Politécnica, Universidade de São Paulo.

Shutt, J.N. (1993). *Recursive adaptable grammar.* M.S. Thesis, Computer Science Department, Worcester Polytechnic Institute, Worcester MA.

KEY TERMS

Adaptivity: Property exhibited by structures that dynamically and autonomously change their own behavior in response to input stimuli.

Adaptive Computation Model: Turing-powerful abstraction that mimic the behavior of potentially self-modifying complex systems.

Adaptive Device: Structure with dynamic behavior, with some subjacent device and an adaptive mechanism.

Adaptive Functions and Adaptive Actions: Adaptive actions are calls to adaptive functions, which can determine changes to perform on its layer's rule set and on their immediately subjacent layer's adaptive functions.

Adaptive Mechanism: Alteration discipline associated to an adaptive device's rule set that change the behavior of its subjacent device by performing adaptive actions.

Adaptive Rule-Driven Device: Adaptive device whose behavior is defined by a dynamically changing set of rules, e.g. adaptive automata, adaptive grammars, etc.

Context-Dependency: Reinterpretation of terms, due to conditions occurring elsewhere in a sentence, e.g. agreement rules in English, type-checking in Pascal.

Context-Sensitive (-Dependent) Formalism: Abstraction capable of representing Chomsky type-1 or type-0 languages. Adaptive Automata and Adaptive Context-free Grammars are well suited to express such languages.

Hierarchical (Multilevel) Adaptive Device: Stratified adaptive structures whose involving layer's adaptive actions can modify both its own layer's rules and its underlying layer's adaptive functions.

Subjacent (or Underlying) Device: Any device used as basis to formulate adaptive devices. The innermost of a multilevel subjacent device must be non-adaptive.

Advanced Cellular Neural Networks Image Processing

J. Álvaro Fernández
University of Extremadura, Badajoz, Spain

INTRODUCTION

Since its introduction to the research community in 1988, the Cellular Neural Network (CNN) (Chua & Yang, 1988) paradigm has become a fruitful soil for engineers and physicists, producing over 1,000 published scientific papers and books in less than 20 years (Chua & Roska, 2002), mostly related to Digital Image Processing (DIP). This Artificial Neural Network (ANN) offers a remarkable ability of integrating complex computing processes into compact, real-time programmable analogic VLSI circuits as the ACE16k (Rodríguez *et al.*, 2004) and, more recently, into FPGA devices (Perko *et al.*, 2000).

CNN is the core of the revolutionary Analogic Cellular Computer (Roska *et al.*, 1999), a programmable system based on the so-called CNN Universal Machine (CNN-UM) (Roska & Chua, 1993). Analogic CNN computers mimic the anatomy and physiology of many sensory and processing biological organs (Chua & Roska, 2002).

This article continues the review started in this Encyclopaedia under the title *Basic Cellular Neural Network Image Processing*.

BACKGROUND

The standard CNN architecture consists of an $M \times N$ rectangular array of cells $C(i,j)$ with Cartesian coordinates (i,j), $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$. Each cell or neuron $C(i,j)$ is bounded to a sphere of influence $S_r(i,j)$ of positive integer radius r, defined by:

$$S_r(i,j) = \left\{ C(k,l) \middle| \max_{1 \le k \le M, 1 \le l \le N} \left\{ |k-i|, |l-j| \right\} \le r \right\}$$

(1)

This set is referred as a $(2r+1) \times (2r+1)$ neighbourhood. The parameter r controls the connectivity

of a cell. When $r > N/2$ and $M = N$, a fully connected CNN is obtained, a case that corresponds to the classic Hopfield ANN model.

The state equation of any cell $C(i,j)$ in the $M \times N$ array structure of the standard CNN may be described by:

$$C \frac{dz_{ij}(t)}{dt} = -\frac{1}{R} z_{ij}(t) + \sum_{C(k,l) \in S_r(i,j)} \left[A(i,j;k,l) \cdot y_{kl}(t) + B(i,j;k,l) \cdot x_{kl} \right] + I_{ij}$$

(2)

where C and R are values that control the transient response of the neuron circuit (just like an RC filter), I is generally a constant value that biases the state matrix $Z = \{z_{ij}\}$, and S_r is the local neighbourhood defined in (**1**), which controls the influence of the input data $X = \{x_{ij}\}$ and the network output $Y = \{y_{ij}\}$ for time t.

This means that both input and output planes interact with the state of a cell through the definition of a set of real-valued weights, $A(i,j;k,l)$ and $B(i,j;k,l)$, whose size is determined by r. The cloning templates A and B are called the feedback and feed-forward operators, respectively.

An isotropic CNN is typically defined with constant values for r, I, A and B, implying that for an input image X, a neuron $C(i,j)$ is provided for each pixel (i,j), with constant weighted circuits defined by the feedback and feed-forward templates A and B. The neuron state value z_{ij} is adjusted with the bias parameter I, and passed as input to an output function of the form:

$$y_{ij} = \frac{1}{2} \left(\left| z_{ij}(t) + 1 \right| - \left| z_{ij}(t) - 1 \right| \right)$$

(3)

The vast majority of the templates defined in the CNN-UM template compendium of (Chua & Roska, 2002) are based on this isotropic scheme, using $r = 1$ and binary images in the input plane. If no feedback (i.e. $A = 0$) is used, then the CNN behaves as a convolution network, using B as a spatial filter, I as a threshold and the piecewise linear output (**3**) as a limiter. Thus,

virtually any spatial filter from DIP theory can be implemented on such a feed-forward CNN, ensuring binary output stability via the definition of a central feedback absolute value greater than 1.

ADVANCED CNN IMAGE PROCESSING

In this section, a description of more complex CNN models is performed in order to provide a deeper insight into CNN design, including multi-layer structures and nonlinear templates, and also to illustrate its powerful DIP capabilities.

Nonlinear Templates

A problem often addressed in DIP edge detection is the robustness against noise (Jain, 1989). In this sense, the EDGE CNN detector for grey-scale images given by

$$A = 2, \; B_{EDGE} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, I = \text{-0.5} \tag{4}$$

is a typical example of a weak-against-noise filter, as a result of fixed linear feed-forward template combined with excitatory feedback. One way to provide the detector with more robustness against noise is via the definition of a nonlinear B template of the form:

$$B_{CONTOUR} = \begin{bmatrix} b & b & b \\ b & 0 & b \\ b & b & b \end{bmatrix} \text{ where } b = \begin{cases} 0.5 & |x_{ij} - x_{kl}| > th \\ -1 & |x_{ij} - x_{kl}| \le th \end{cases} \tag{5}$$

This nonlinear template actually defines different coefficients for the surrounding pixels prior to perform the spatial filtering of the input image X. Thus, a CNN defined with nonlinear templates is generally dependent of X, and can not be treated as an isotropic model.

Just two values for the surrounding coefficients of B are allowed: one excitatory for greater than a threshold th luminance differences with the central pixel (i.e. edge pixels), and the other inhibitory, doubled in absolute value, for similar pixels, where th is usually set around

0.5. The feedback template $A = 2$ remains unchanged, but the value for the bias I must be chosen from the following analysis:

For a given state z_{ij} element, the contribution w_{ij} of the feed-forward nonlinear filter of (**5**) may be expressed as:

$$\begin{aligned} w_{ij} &= -1.0 \cdot p_s + 0.5 \cdot p_e \\ &= -(8 - p_e) + 0.5 \cdot p_e \\ &= -8 + 1.5 \cdot p_e \end{aligned} \tag{6}$$

where p_s is the number of similar pixels in the 3×3 neighbourhood and p_e the rest of edge pixels. E.g. if the central pixel has 8 edge neighbours, $w_{ij} = 12 - 8 = 4$, whereas if all its neighbours are similar to it, then $w_{ij} = -8$. Thus, a pixel will be selected as edge depending on the number of its edge neighbours, providing the possibility of noise reduction. For instance, edge detection for pixels with at least 3 edge neighbours forces that $I \in (4, 5)$.

The main result is that the inclusion of nonlinearities in the definition of B coefficients and, by extension, the pixel-wise definition of the main CNN parameters gives rise to more powerful and complex DIP filters (Chua & Roska, 1993).

Morphologic Operators

Mathematical Morphology is an important contributor to the DIP field. In the classic approach, every morphologic operator is based on a series of simple concepts from Set Theory. Moreover, all of them can be divided into combinations of two basic operators: erosion and dilation (Serra, 1982). Both operators take two pieces of data as input: the binary input image and the so-called structuring element, which is usually represented by a 3×3 template.

A pixel belongs to an object if it is active (i.e. its value is 1 or black), whereas the rest of pixels are classified as background, zero-valued elements. Basic morphologic operators are defined using only object pixels, marked as 1 in the structuring element. If a pixel is not used in the match, it is left blank. Both dilation and erosion operators may be defined by the structuring elements

1	1	1
1	1	1
1	1	1

	1	
1	1	1
	1	

and $\qquad$ (7)

for 8 or 4-neighbour connectivity, respectively. In dilation, the structuring element is placed over each input pixel. If any of the 9 (or 5) pixels considered in (**7**) is active, then the output pixel will be also active (Jain, 1989). The erosion operator can be defined as the dual of dilation, i.e. a dilation performed over the background.

More complex morphologic operators are based on structuring elements that also contains background pixels. This is the case of the Hit and Miss Transform (HMT), a generalized morphologic operator used to identify certain local pixel configurations. For instance, the structuring elements defined by

0	1	
0	1	1
0	0	0

1		1
0	1	0
0	0	0

and $\qquad$ (8)

are used to find 90º convex corner object pixels within the image. A pixel will be selected as active in the output image if its local neighbourhood exactly matches with that defined by the structuring element. However, in order to calculate a full, non-orientated corner detector it will be necessary to perform 8 HMT, one for each rotated version of (**8**), OR-ing the 8 intermediate output images to obtain the final image (Fisher *et al.*, 2004).

In the CNN context, the HMT may be obtained in a straightforward manner by:

$$A = 2, \ B_{HMT} : b_{ij} = \begin{cases} 1 & s_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}, \ I = 0.5 - p_s$$

(9)

where $S = \{s_{ij}\}$ is the structuring element and p_s is the total number of active pixels in it.

Since the input template B of the HTM CNN is defined via the structuring element S, and given that there are $2^9 = 512$ distinct 3×3 possible structuring elements, there will also be 512 different hit-and-miss erosions. For achieving the opposite result, i.e. hit-and-miss dilation, the threshold must be the opposite of that in (**9**) (Chua & Roska, 2002).

Dynamic Range Control CNN and Piecewise Linear Mappings

A

DIP techniques can be classified by the domain where they operate: the image or spatial domain or the transform domain (e.g. the Fourier domain). Spatial domain techniques are those who operate directly over the pixels within an image (e.g. its intensity level). A generic spatial operator can be defined by

$$Y(i, j) = T \left[X(i, j) \right]_{S_r}$$

(10)

where X and Y are the input and output images, respectively, and T is a spatial operator defined over a neighbourhood S_r around each pixel $X(i, j)$, as defined in (**1**). Based on this neighbourhood, spatial operators can be grouped into two types: Single Point Processing Operators, also known as Mapping Operators, and Local Processing Operators, which can be defined by a spatial filter (i.e. 2D-discrete convolution) mask (Jain, 1989).

The simplest form of T is obtained when S_r is 1 pixel size. In this case, Y only depends of the intensity value of X for every pixel and T becomes an intensity level transformation function, or mapping, of the form

$$s = T(r)$$

(11)

where r and s are variables that represent grey level in X and Y, respectively.

According to this formulation, mappings can be achieved by direct application of a function over a range of input intensity levels. By properly choosing the form of T, a number of effects can be obtained, as the grey-level inversion, dynamic range compression or expansion (i.e. contrast enhancement), and threshold binarization for obtaining binary masks used in analysis and morphologic DIP.

A mapping is linear if its function T is also linear. Otherwise, T is not linear and the mapping is also nonlinear. An example of nonlinear mapping is the CNN output function (**3**). It consists of three linear segments: two saturated levels, –1 and +1, and the central linear segment with unitary slope that connects them. This function is said to be piecewise linear and is closely related to the well-known sigmoid function utilized in the Hopfield ANN (Chua & Roska, 1993). It performs a mapping of intensity values stored in Z in the [–1,

+1] range. The bias *I* controls the average point of the input range, where the output function gives a zero-valued outcome.

Starting from the original CNN cell or neuron (**1**)-(**3**), a brief review of the Dynamic Range Control (DRC) CNN model first defined in (Fernández *et al.*, 2006) follows. This network is designed to perform a piecewise linear mapping *T* over *X*, with input range [*m–d*, *m+d*] and output range [*a*, *b*]. Thus,

$$T\left[X\left(i,j\right)\right] = \begin{cases} a & -\infty < X\left(i,j\right) \le m-d \\ \dfrac{b-a}{2d}\left(X\left(i,j\right)-m\right)+\dfrac{b+a}{2} & m-d < X\left(i,j\right) \le m+d \\ b & m+d < X\left(i,j\right) < +\infty \end{cases}$$

$$(12)$$

In order to be able to implement this function in a multi-layer CNN, the following constraints must be met:

$$|b-a| \le 2 \text{ and } d \le 1 \qquad (13)$$

A CNN cell which controls the desired input range can be defined with the following parameters:

$$A_1 = 0, B_1 = 1/d, I_1 = -m/d \qquad (14)$$

This network performs a linear mapping between [*m–d*, *m+d*] and [–1,+1]. Its output is the input of a second CNN whose parameters are:

$$A_2 = 0, B_2 = (b-a)/2, I_2 = (b+a)/2 \qquad (15)$$

The output of this second network is exactly the mapping *T* defined in (**12**) bounded by the constraints of (**13**).

One of the simplest techniques used in grey-scale image contrast enhancement is contrast stretching or normalization. This technique maximizes the dynamic range of the intensity levels within the image from suitable estimates of the maximum and minimum intensity values (Fisher *et al.*, 2004). Thus, in the case of normalized grey-scale images, where the minimum (i.e. black) and maximum (i.e. white) intensity levels are represented by 0 and 1 values, respectively; if such an image with dynamic intensity range [*f*, *g*] ⊆ [0, +1] is fed in the input of the 2-layer CNN defined by (**14**) and (**15**), the following parameters will achieve the desired linear dynamic range maximization:

$$a = 0, b = 1, m = (g + f)/2, d = (g - f)/2$$

$$(16)$$

The DRC network can be easily applied to a first order piecewise polynomial approximation of nonlinear, continuous mappings. One of the valid possibilities is the multi-layer DRC CNN implementation of error-controlled Chebyshev polynomials, as described in (Fernández *et al.*, 2006). The possible mappings include, among many others, the absolute value, logarithmic, exponential, radial basis and integer and real-valued power functions.

FUTURE TRENDS

There is a continuous quest by engineers and specialists: compete with and imitate nature, especially some "smart" animals. Vision is one particular area which computer engineers are interested in. In this context, the so-called Bionic Eye (Werblin *et al.*, 1995) embedded in the CNN-UM architecture is ideal for implementing many spatio-temporal neuromorphic models.

With its powerful image processing toolbox and a compact VLSI implementation (Rodríguez *et al.*, 2004), the CNN-UM can be used to program or mimic different models of retinas and even combinations of them. Moreover, it can combine biologically based models, biologically inspired models, and analogic artificial image processing algorithms. This combination will surely bring a broader kind of applications and developments.

CONCLUSION

A number of other advances in the definition and characterization of CNN have been researched in the past decade. This includes the definition of methods for designing and implementing larger than 3×3 neighbourhoods in the CNN-UM (Kék & Zarándy, 1998), the CNN implementation of some image compression techniques (Venetianer *et al.*, 1995) or the design of a CNN-based Fast Fourier Transform algorithm over analogic signals (Perko *et al.*, 1998), between many others.

In this article, a general review of the main properties and features of the Cellular Neural Network model has been addressed focusing on its DIP applications. The

CNN is now a fundamental and powerful toolkit for real-time nonlinear image processing tasks, mainly due to its versatile programmability, which has powered its hardware development for visual sensing applications (Roska *et al.*, 1999).

REFERENCES

Chua, L.O., & Roska, T. (2002). *Cellular Neural Networks and Visual Computing. Foundations and Applications*. Cambridge, UK: Cambridge University Press.

Chua, L.O., & Roska, T. (1993). The CNN Paradigm. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 40, 147–156.

Chua, L.O., & Yang, L. (1988). Cellular Neural Networks: Theory and Applications. *IEEE Transactions on Circuits and Systems*, 35, 1257–1290.

Fernández, J.A., Preciado, V.M., & Jaramillo, M.A. (2006). Nonlinear Mappings with Cellular Neural Networks. *Lecture Notes in Computer Science*, 4177, 350–359.

Fisher, R., Perkins, S., Walker, A., & Wolfart, E. (2004). *Hypermedia Image Processing Reference (HIPR2)*. Website: http://homepages.inf.ed.ac.uk/rbf/HIPR2, University of Edinburgh, UK.

Jain, A.K. (1989). *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall.

Kék, L., & Zarándy, A. (1998). Implementation of Large Neighborhood Non-Linear Templates on the CNN Universal Machine. *International Journal of Circuit Theory and Applications,* 26, 551-566.

Perko, M., Fajfar, I., Tuma, T., & Puhan, J. (1998). Fast Fourier Transform Computation Using a Digital CNN Simulator. *5th IEEE International Workshop on Cellular Neural Network and Their Applications Proceedings*, 230-236.

Perko, M., Fajfar, I., Tuma, T., & Puhan, J. (2000). Low-Cost, High-Performance CNN Simulator Implemented in FPGA. *6th IEEE International Workshop on Cellular Neural Network and Their Applications Proceedings*, 277-282.

Rodríguez, A., Liñán, G., Carranza, L., Roca, E., Carmona, R., Jiménez, F., Domínguez, R., & Espejo, S. (2004). ACE16k: The Third Generation of Mixed-Signal SIMD-CNN ACE Chips Toward VSoCs. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51, 851–863.

Roska, T., & Chua, L.O. (1993). The CNN Universal Machine: An Analogic Array Computer. *IEEE Transactions on Circuits and Systems II: Analog and Digital Processing*, 40, 163–173.

Roska, T., Zarándy, Á., Zöld, S., Földesy, P., & Szolgay, P. (1999). The Computational Infrastructure of Analogic CNN Computing – Part I: The CNN-UM Chip Prototyping System. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 46, 261–268.

Serra, J. (1982). *Image Analysis and Mathematical Morphology. London, UK:* Academic Press.

Venetianer, P.L., Werblin, F., Roska, T., & Chua, L.O. (1995). Analogic CNN Algorithms for Some Image Compression and Restoration Tasks. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* 42, 278-284.

Werblin, F., Roska, T., & Chua, L.O. (1995). The Analogic Cellular Neural Network as a Bionic Eye. *International Journal of Circuit Theory and Applications,* 23, 541-569.

KEY TERMS

Bionics: The application of methods and systems found in nature to the study and design of engineering systems. The word seems to have been formed from "**bio**logy" and "electro**nics**" and was first used by J. E. Steele in 1958.

Chebyshev Polynomial: An important type of polynomials used in data interpolation, providing the best approximation of a continuous function under the maximum norm.

Dynamic Range: A term used to describe the ratio between the smallest and largest possible values of a variable quantity.

FPGA: Acronym that stands for Field-Programmable Gate Array, a semiconductor device invented

A

in 1984 by R. Freeman that contains programmable interfaces and logic components called "logic blocks" used to perform the function of basic logic gates (e.g. XOR) or more complex combination functions such as decoders.

Piecewise Linear Function: A function $f(x)$ that can be split into a number of linear segments, each of which is defined for a non-overlapping interval of x.

Spatial Convolution: A term used to identify the linear combination of a series of discrete 2D data (a digital image) with a few coefficients or weights. In the Fourier theory, a convolution in space is equivalent to (spatial) frequency filtering.

Template: Also known as kernel, or convolution kernel, is the set of coefficients used to perform a spatial filter operation over a digital image via the spatial convolution operator.

VLSI: Acronym that stands for Very Large Scale Integration. It is the process of creating integrated circuits by combining thousands (nowadays hundreds of millions) of transistor-based circuits into a single chip. A typical VLSI device is the microprocessor.

Agent–Based Intelligent System Modeling

Zaiyong Tang
Salem State College, USA

Xiaoyu Huang
University of Shanghai for Science & Technology, China

Kallol Bagchi
University of Texas at El Paso, USA

INTRODUCTION

An intelligent system is a system that has, similar to a living organism, a coherent set of components and subsystems working together to engage in goal-driven activities. In general, an intelligent system is able to sense and respond to the changing environment; gather and store information in its memory; learn from earlier experiences; adapt its behaviors to meet new challenges; and achieve its pre-determined or evolving objectives. The system may start with a set of predefined stimulus-response rules. Those rules may be revised and improved through learning. Anytime the system encounters a situation, it evaluates and selects the most appropriate rules from its memory to act upon.

Most human organizations such as nations, governments, universities, and business firms, can be considered as intelligent systems. In recent years, researchers have developed frameworks for building organizations around intelligence, as opposed to traditional approaches that focus on products, processes, or functions (e.g., Liang, 2002; Gupta and Sharma, 2004). Today's organizations must go beyond traditional goals of efficiency and effectiveness; they need to have organizational intelligence in order to adapt and survive in a continuously changing environment (Liebowitz, 1999). The intelligent behaviors of those organizations include monitoring of operations, listening and responding to stakeholders, watching the markets, gathering and analyzing data, creating and disseminating knowledge, learning, and effective decision making.

Modeling intelligent systems has been a challenge for researchers. Intelligent systems, in particular, those involve multiple intelligent players, are complex systems where system dynamics does not follow clearly defined rules. Traditional system dynamics approaches or statistical modeling approaches rely on rather restrictive assumptions such as homogeneity of individuals in the system. Many complex systems have components or units which are also complex systems. This fact has significantly increased the difficulty of modeling intelligent systems. Agent-based modeling of complex systems such as ecological systems, stock market, and disaster recovery has recently garnered significant research interest from a wide spectrum of fields from politics, economics, sociology, mathematics, computer science, management, to information systems. Agent-based modeling is well suited for intelligent systems research as it offers a platform to study systems behavior based on individual actions and interactions. In the following, we present the concepts and illustrate how intelligent agents can be used in modeling intelligent systems.

We start with basic concepts of intelligent agents. Then we define agent-based modeling (ABM) and discuss strengths and weaknesses of ABM. The next section applies ABM to intelligent system modeling. We use an example of technology diffusion for illustration. Research issues and directions are discussed next, followed by conclusions.

INTELLIGENT AGENT

Intelligent agents, also known as software agents, are computer applications that autonomously sense and respond to environment in the pursuit of certain designed objectives (Wooldridge and Jennings, 1995). Intelligent agents exhibit some level of intelligence. They can be

used to assist the user in performing non-repetitive tasks, such as seeking information, shopping, scheduling, monitoring, control, negotiation, and bargaining.

Intelligent agents may come in various shapes and forms such as knowbots, softbots, taskbots, personal agents, shopbots, information agents, etc. No matter what shape or form they have, intelligent agents exhibit one or more of the following characteristics:

- **Autonomous:** Being able to exercise control over their own actions.
- **Adaptive/Learning:** Being able to learn and adapt to their external environment.
- **Social:** Being able to communicate, bargain, collaborate, and compete with other agents on behalf of their masters (users).
- **Mobile:** Being able to migrate themselves from one machine/system to another in a network, such as the Web.
- **Goal-oriented:** Being able to act in accordance with built-in goals and objectives.
- **Communicative:** Being able to communicate with people or other agents thought protocols such as agent communication language (ACL).
- **Intelligent:** Being able to exhibit intelligent behavior such as reasoning, generalizing, learning, dealing with uncertainty, using heuristics, and natural language processing.

AGENT-BASED MODELING

Using intelligent agents and their actions and interactions in a given environment to simulate the complex dynamics of a system is referred to as agent-based modeling. ABM research is closely related to the research in complex systems, emergence, computational sociology, multi agent systems, evolutionary programming, and intelligent organizations. In ABM, system behavior results from individual behaviors and collective behaviors of the agents. Researchers of ABM are interested in how macro phenomena are emerging from micro level behaviors among a heterogeneous set of interacting agents (Holland, 1992). Every agent has its attributes and its behavior rules. When agents encounter in the agent society, each agent individually assesses the situation and makes decisions on the basis of its behavior rules. In general, individual agents do

not have global awareness in the multi-agent system.

Agent-based modeling allows a researcher to set different parameters and behavior rules of individual agents. The modeler makes assumptions that are most relevant to the situation at hand, and then watches phenomena emerge from the interactions of the agents. Various hypotheses can be tested by changing agent parameters and rules. The emergent collective pattern of the agent society often leads to results that may not have been predicated.

One of the main advantages of ABM over traditional mathematical equation based modeling is the ability to model individual styles and attributes, rather than assuming homogeneity of the whole population. Traditional models based on analytical techniques often become intractable as the systems reach real-world level of complexity. ABM is particularly suitable for studying system dynamics that are generated from interactions of heterogeneous individuals. In recent years, ABM has been used in studying many real world systems, such as stock markets (Castiglione 2000), group selection (Pepper 2000), and workflow and information diffusion (Neri 2004). Bonabeau (2002) presents a good summary of ABM methodology and the scenarios where ABM is appropriate.

ABM is, however, not immune from criticism. Per Bonabeau (2002), "an agent-based model will only be as accurate as the assumptions and data that went into it, but even approximate simulations can be very valuable". It has also been observed that ABM relies on simplified models of rule-based human behavior that often fail to take into consideration the complexity of human cognition. Besides, it suffers from "unwrapping" problem as the solution is built into the program and thus prevents occurrence of new or unexpected events (Macy, 2002).

ABM FOR INTELLIGENT SYSTEMS

An intelligent system is a system that can sense and respond to its environment in pursuing its goals and objectives. It can learn and adapt based on past experience. Examples of intelligent systems include, but not limited to, the following: biological life such as human beings, artificial intelligence applications, robots, organizations, nations, projects, and social movements.

Walter Fritz (1997) suggests that the key components of an intelligent system include objectives, senses, concepts, growth of a concept, present situation, response rules, mental methods, selection, actions, reinforcement, memory and forgetting, sleeping, and patterns (high level concepts). It is apparent that traditional analytical modeling techniques are not able to model many of the components of intelligent systems, let alone the complete system dynamics. However, ABM lends itself well to such a task. All those components can be models as agents (albeit some in abstract sense). An intelligent system is thus made of inter-related and interactive agents. ABM is especially suitable for intelligent systems consist of a large number of heterogeneous participants, such as a human organization.

Modeling Processes

Agent-based modeling for intelligent systems starts with a thorough analysis of the intelligent systems. Since the system under consideration may exhibit complex behaviors, we need to identify one or a few key features to focus on. Given a scenario of the target intelligent system, we first establish a set of objectives that we aim to achieve via the simulation of the agent-based representation of the intelligent system. The objectives of the research can be expressed as a set of questions to which we seek answers (Doran, 2006).

A conceptual model is created to lay out the requirements for achieving the objectives. This includes defining the entities, such as agents, environment, resources, processes, and relationships. The conceptual modeling phase answers the question of what—what are needed. The design model determines how the requirements can be implemented, including defining the features and relevant behaviors of the agents (Brown, 2006).

Depending on the goals of a particular research, a model may involve the use of designed or empirically grounded agents. Designed agents are those endowed with characteristics and behaviors that represent conditions for testing specific hypotheses about the intelligent systems. When the agents are empirically grounded, they are used to represent real world entities, such as individuals or processes in an organization. Empirically grounded agents are feasible only when data about the real world entities are available. Similarly, the environment within which the agents act can be

designed or empirically grounded. In practice, a study may start with simple models, often with designed agents and environments, to explore certain specific dynamics of the system.

The design model is refined through the calibration process, in which design parameters are modified to improve the desired characteristics of the model. The final step in the modeling process is validation where we check the agent individual behavior, interactions, and emergent properties of the system against expected design features. Validation usually involves comparison of model outcomes, often at the macro-level, with comparable outcomes in the real world (Midgley, el at., 2007). Figure 1 shows the complete modeling process. A general tutorial on ABM is given by Macal and North (2005).

ABM for Innovation Diffusion

We present an example of using agent-based intelligent system modeling for studying the acceptance and diffusion of innovative ideas or technology. Diffusion of innovation has been studied extensively over the last few decades (Rogers, 1995). However, traditional research in innovation diffusion has been grounded on case based analysis and analytical systems modeling

Figure 1. Agent-based modeling process
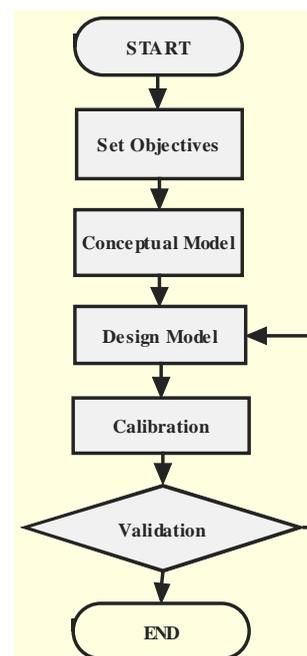
(e.g., using differential and difference equations). Agent-based modeling for diffusion of innovation is relatively new. Our example is adopted from a model created by Michael Samuels (2007), implemented with a popular agent modeling system—NetLogo.

The objective of innovation diffusion modeling is to answer questions such as how an idea or technology is adopted in a population, how different people (e.g., innovators, early adopters, and change agents) influence each other, and under what condition an innovation will be accepted or rejected by the population. In the conceptual modeling, we identify various factors that influence an individual's propensity for adopting the innovation. Those factors are broadly divided into to two categories: internal influences (e.g., word-of-mouth) and external influences (e.g. mass media). Any factor that exerts its influence through individual contact is considered internal influence.

Individuals in the target population are divided into four groups: *adopter*, *potential* (adopter), *change agent*, and *disrupter*. *Adopters* are those who have adopted the innovation, while *potentials* are those who have certain likelihood to adopt the innovation. *Change agents* are the champions of the innovation. They are very knowledgeable and enthusiastic about the innovation, and often play a critical role in facilitating its-diffusion. *Disrupters* are those who play an opposite role of *change agents.* They are against the current innovation, oftentimes because they favor an even

newer and perceived better innovation. The four groups of agents and their relationships are depicted in Figure 2. It is common, although not necessary, to assume that those four groups make up the entire population.

In a traditional diffusion model, such as the Bass model (Bass, 1996), the diffusion rate depends only on the number of adopters (and potential adopters, given fixed population size). Characteristics of individuals in the population are ignored. Even in those models where it is assumed that potential adopters have varying threshold for adopting an innovation (Abrahamson and Rosenkopf, 1997), the individuality is very limited. However, in agent-based modeling, the types of individuals and individual characteristics are essentially unbounded. For example, we can divide easily adopters into innovators, early adopters, and late adopters, etc. If necessary, various demographic and social-economic features can be bestowed to individual agents. Furthermore, both internal influence and external influence can be further attributed to more specific causes. For example, internal influence through social networks can be divided into traditional social networks that consists friends and acquaintances and virtual social networks formed online. Table 1 lists typical factors that affect the propensity of adopting an innovation.

An initial study of innovation diffusion, such as the one in Michael Samuels (2007), can simply aggregate all internal influences into "word-of-month" and all external influences into mass media. Each potential adopter's tendency of converting to an adopter is influenced by chance encounter with other agents. If a potential adopter meets a change agent, who is an avid promoter of the innovation, he would become more knowledgeable about the advantages of the innovation, and more likely to adopt. An encounter with a disrupter

Figure 2. Agents and influences

Dotted line:External influence
Solid line: Internal influence
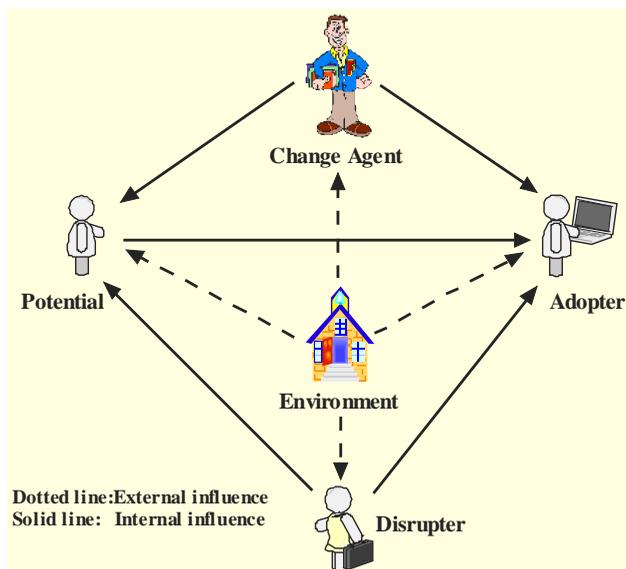
Table1. Typical internal and external influences

Internal influence	External influence
Word-of-mouth	Newspapers
Telephone	Television
Email	Laws, policies and regulations
Instant message	Culture
Chat	Internet/Web
Blog	Online communities
Social networks (online/ offline)	RSS

creates the opposite effect, as a disrupter favors a different type of innovation.

In order for the simulated model to accurately reflect a real-world situation, the model structure and parameter values should be carefully selected. For example, we need to decide how much influence each encounter will result; what is the probability of encountering a change agent or a disrupter; how much influence is coming from the mass media, etc. We can get these values through surveys, statistical analysis of empirical data, or experiments specifically designed to elicit data from real world situations.

TRENDS AND RESEARCH ISSUES

As illustrated through the example of modeling the diffusion of innovation in an organization, industry, or society, agent-based modeling can be used to model the adaptation of intelligent systems that consist of intelligent individuals. As most intelligent systems are complex in both structure and system dynamics, traditional modeling tools that require too many unrealistic assumptions have become less effective in modeling intelligent systems. In recent years, agent-based modeling has found a wide spectrum of applications such as in business strategic solutions, supply chain management, stock markets, power economy, social evolution, military operations, security, and ecology (North and Macal, 2007). As ABM tools and resources become more accessible, research and applications of agent-based intelligent system modeling are expected to increase in the near future.

Some challenges remain, though. Using ABM to model intelligent systems is a research area that draws theories from other fields, such as economics, psychology, sociology, etc., but without its own well established theoretic foundation. ABM has four key assumptions (Macy and Willer, 2002): Agents act locally with little or no central authority; agents are interdependent; agents follow simple rules, and agents are adaptive. However, some of those assumptions may not be applicable to intelligent system modeling. Central authorities, or central authoritative information such as mass media in the innovation diffusion example, may play an important role in intelligent organizations. Not all agents are alike in an intelligent system. Some may be independent, non-adaptive, or following complex behavior rules.

ABM uses a "bottom-up" approach, creating emergent behaviors of an intelligent system through "actors" rather than "factors". However, macro-level factors have direct impact on macro behaviors of the system. Macy and Willer (2002) suggest that bringing those macro-level factors back will make agent-based modeling more effective, especially in intelligent systems such as social organizations.

Recent intelligent systems research has developed the concept of integrating human and machine-based data, knowledge, and intelligence. Kirn (1996) postulates that the organization of the 21st century will involve artificial agents based system highly intertwined with human intelligence of the organization. Thus, a new challenge for agent-based intelligent system modeling is to develop models that account for interaction, aggregation, and coordination of intelligent agent and human agents. The ABM will represent not only the human players in an intelligent system, but also the intelligent agents that are developed in real-world applications in those systems.

CONCLUSION

Modeling intelligent systems involving multiple intelligent players has been difficult using traditional approaches. We have reviewed recent development in agent-based modeling and suggest agent-based modeling is well suited for studying intelligent systems, especially those systems with sophisticated and heterogeneous participants. Agent-based modeling allows us to model system behaviors based on the actions and interactions of individuals in the system. Although most ABM research focuses on local rules and behaviors, it is possible that we integrate global influences in the models. ABM represents a novel approach to model intelligent systems. Combined with traditional modeling approaches (for example, micro-level simulation as proposed in MoSeS), ABM offers researchers a promising tool to solve complex and practical problems and to broaden research endeavors (Wu, 2007).

REFERENCES

Abrahamson, E. and L. Rosenkopf (1997). Social Network Effects on the Extent of Innovation Diffusion: A Computer Simulation. Organization Science. 8(3), 289-309.

Bass, F. M. (1969). A New Product Growth Model for Consumer Durables, Management Science, 13(5). 215-227.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. PNAS May 14, 2002. 99, suppl. 3, 7280-7287.

Brown, D.G. (2006). Agent-based models. In H. Geist, Ed. The Earth's Changing Land: An Encyclopedia of Land-Use and Land-Cover Change. Westport CT: Greenwood Publishing Group. 7-13.

Doran J. E. (2006). Agent Design for Agent Based Modeling. In Agent Based Computational Modelling: Applications in Demography, Social, Economic and Environmental Sciences, eds. F. C. Billari, T. Fent, A. Prskawetz, and J.Scheffran. Physica-Verlag (Springer). 215-223.

Filippo Castiglione (2000), 'Diffusion and aggregation in an agent based model of stock market fluctuations', International Journal of Modern Physics C. 11(5), 1-15.

Fritz, Walter (1997).Intelligent Systems and their Societies. First version: Jan 27, 1997 http://www. intelligent-systems.com.ar/intsyst/index.htm

Gupta, J. N. D. and S. K. Sharma (2004). Editors. Intelligent Enterprises for the 21st Century. Hershey, PA: Idea Group Publishing.

Holland, J.H. (1992). Complex adaptive systems. Daedalus. 121(1), 17-30.

Kirn, S. 1996. Organizational intelligence and distributed artificial intelligence. In *Foundations of Distributed Artificial intelligence*, G. M. O'Hare and N. R. Jennings, Eds. John Wiley Sixth-Generation Computer Technology Series. John Wiley & Sons, New York, NY. 505-526.

Liang, T. Y. (2002). The Inherent Structure and Dynamic of Intelligent Human Organizations, Human Systems Management. 21(1), 9-19.

Liebowitz, J. (1999). Building Organizational Intelligence: A Knowledge Primer, New York: CRC Press.

Macal, C. M. and North, M. J. (2005). Tutorial on Agent-Based Modeling and Simulation. Proceedings of the 37th Winter Simulation Conference, Orlando, Florida. 2-15.

Macy, M. W. (2002). Social Simulation, In N. Smelser and P. Baltes, eds., International Encyclopedia of the Social and Behavioral Sciences, Elsevier, The Netherlands.

Macy, M.W, and Willer, R. (2002). From Factors to Actors: Computational Sociology and Agent-Based Modeling. Annual Review of Sociology. 28, 143-166.

McMaster, M. D. (1996). The Intelligence Advantage: Organizing for Complexity. Burlington MA: Butterworth-Heineman.

Midgley, D.F., Marks R.E., and Kunchamwar D. (2007). The Building and Assurance of Agent-Based Models: An Example and Challenge to the Field. Journal of Business Research. 60(8), 884-893.

Neri, F. (2004). Agent Based Simulation of Information Diffusion in a Virtual Market Place. IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04). 333-336.

North, M. J. and C. M. Macal, (2007). Managing Business Complexity: Discovering Strategic Solutions with Agent-based Modeling and Simulation. Oxford University Press, New York.

Pepper, J. W. (2000) An Agent-Based Model of Group Selection, Santa Fe Institute. Retrieved June 16, 2007 at: http://www.santafe.edu/~jpepper/papers/ALIFE7_GS.pdf

Rogers, E.M. (1995). Diffusion of Innovations. The Free Press, New York.

Samuels, M.L. (2007). Innovation model. Last updated: 01/08/2007, http://ccl.northwestern.edu/netlogo/models/community/Innovation.

Wooldridge, M. and N. R. Jennings (1995). Intelligent Agents: Theory and Practice, Knowledge Engineering Review. 10(2), 115-152.

Wu, B. (2007). A Hybrid Approach for Spatial MSM. NSF/ESRC Agenda Setting Workshop on Agent-Based Modeling of Complex Spatial Systems: April 14-16, 2007

KEY TERMS

Agent Based Modeling: Using intelligent agents and their actions and interactions in a given environment to simulate the complex dynamics of a system.

Diffusion of Innovation: Popularized by Everett Rogers, it is the study of the process by which an innovation is communicated and adopted over time among the members of a social system.

Intelligent Agent: An autonomous software program that is able to learn and adapt to its environment in order to perform certain tasks delegated to it by its master.

Intelligent System: A system that has a coherent set of components and subsystems working together to engage in goal-driven activities.

Intelligent System Modeling: The process of construction, calibration, and validation of models of intelligent systems.

Multi-Agent System: A distributed system with a group of intelligent agents that communicate, bargain, compete, and cooperate with other agents and the environment to achieve goals designated by their masters.

Organizational Intelligence: The ability of an organization to perceive, interpret, and select the most appropriate response to the environment in order to advance its goals.

AI and Ideas by Statistical Mechanics

Lester Ingber
Lester Ingber Research, USA

INTRODUCTION

A briefing (Allen, 2004) demonstrates the breadth and depth complexity required to address real diplomatic, information, military, economic (DIME) factors for the propagation/evolution of ideas through defined populations. An open mind would conclude that it is possible that multiple approaches may be required for multiple decision makers in multiple scenarios. However, it is in the interests of multiple decision-makers to as much as possible rely on the same generic model for actual computations. Many users would have to trust that the coded model is faithful to process their inputs.

Similar to DIME scenarios, sophisticated competitive marketing requires assessments of responses of populations to new products.

Many large financial institutions are now trading at speeds barely limited by the speed of light. They co-locate their servers close to exchange floors to be able to turn quotes into orders to be executed within msecs. Clearly, trading at these speeds require automated algorithms for processing and making decisions. These algorithms are based on "technical" information derived from price, volume and quote (Level II) information. The next big hurdle to automated trading is to turn "fundamental" information into technical indicators, e.g., to include new political and economic news into such algorithms.

BACKGROUND

The concept of "memes" is an example of an approach to deal with DIME factors (Situngkir, 2004). The meme approach, using a reductionist philosophy of evolution among genes, is reasonably contrasted to approaches emphasizing the need to include relatively global influences of evolution (Thurtle, 2006).

There are multiple other alternative works being conducted world-wide that must be at least kept in mind while developing and testing models of evolution/propagation of ideas in defined populations: A

study on a simple algebraic model of opinion formation concluded that the only final opinions are extremal ones (Aletti et al., 2006). A study of the influence on chaos on opinion formation, using a simple algebraic model, concluded that contrarian opinion could persist and be crucial in close elections, albeit the authors were careful to note that most real populations probably do not support chaos (Borghesi & Galam, 2006). A limited review of work in social networks illustrates that there are about as many phenomena to be explored as there are disciplines ready to apply their network models (Sen, 2006).

Statistical Mechanics of Neocortical Interactions (SMNI)

A class of AI algorithms that has not yet been developed in this context takes advantage of information known about real neocortex. It seems appropriate to base an approach for propagation of ideas on the only system so far demonstrated to develop and nurture ideas, i.e., the neocortical brain. A statistical mechanical model of neocortical interactions, developed by the author and tested successfully in describing short-term memory (STM) and electroencephalography (EEG) indicators, is the proposed bottom-up model. Ideas by Statistical Mechanics (ISM) is a generic program to model evolution and propagation of ideas/patterns throughout populations subjected to endogenous and exogenous interactions (Ingber, 2006). ISM develops subsets of macrocolumnar activity of multivariate stochastic descriptions of defined populations, with macrocolumns defined by their local parameters within specific regions and with parameterized endogenous inter-regional and exogenous external connectivities. Parameters of subsets of macrocolumns will be fit to patterns representing ideas. Parameters of external and inter-regional interactions will be determined that promote or inhibit the spread of these ideas. Fitting such nonlinear systems requires the use of sampling techniques.

The author's approach uses guidance from his statistical mechanics of neocortical interactions (SMNI),

developed in a series of about 30 published papers from 1981-2001 (Ingber, 1983; Ingber, 1985; Ingber, 1992; Ingber, 1994; Ingber, 1995; Ingber, 1997). These papers also address long-standing issues of information measured by electroencephalography (EEG) as arising from bottom-up local interactions of clusters of thousands to tens of thousands of neurons interacting via short-ranged fibers), or top-down influences of global interactions (mediated by long-ranged myelinated fibers). SMNI does this by including both local and global interactions as being necessary to develop neocortical circuitry.

Statistical Mechanics of Financial Markets (SMFM)

Tools of financial risk management, developed to process correlated multivariate systems with differing non-Gaussian distributions using modern copula analysis enables bona fide correlations and uncertainties of success and failure to be calculated. Since 1984, the author has published about 20 papers developing a Statistical Mechanics of Financial Markets (SMFM), many available at http://www.ingber.com. These are relevant to ISM, to properly deal with real-world distributions that arise in such varied contexts.

Gaussian copulas are developed in a project Trading in Risk Dimensions (TRD) (Ingber, 2006). Other copula distributions are possible, e.g., Student-t distributions. These alternative distributions can be quite slow because inverse transformations typically are not as quick as for the present distribution. Copulas are cited as an important component of risk management not yet widely used by risk management practitioners (Blanco, 2005).

Sampling Tools

Computational approaches developed to process different approaches to modeling phenomena must not be confused with the models of these phenomena. For example, the meme approach lends it self well to a computational scheme in the spirit of genetic algorithms (GA). The cost/objective function that describes the phenomena of course could be processed by any other sampling technique such as simulated annealing (SA). One comparison (Ingber & Rosen, 1992) demonstrated the superiority of SA over GA on cost/objective functions used in a GA database. That study used Very Fast

Simulated Annealing (VFSR), created by the author for military simulation studies (Ingber, 1989), which has evolved into Adaptive Simulated Annealing (ASA) (Ingber, 1993). However, it is the author's experience that the Art and Science of sampling complex systems requires tuning expertise of the researcher as well as good codes, and GA or SA likely would do as well on cost functions for this study.

If there are not analytic or relatively standard math functions for the transformations required, then these transformations must be performed explicitly numerically in code such as TRD. Then, the ASA_PARALLEL OPTIONS already existing in ASA (developed as part of the 1994 National Science Foundation Parallelizing ASA and PATHINT Project (PAPP)) would be very useful to speed up real time calculations (Ingber, 1993). Below, only a few topics relevant to ISM are discussed. More details are in a previous report (Ingber, 2006).

SMNI AND SMFM APPLIED TO ARTIFICIAL INTELLIGENCE

Neocortex has evolved to use minicolumns of neurons interacting via short-ranged interactions in macrocolumns, and interacting via long-ranged interactions across regions of macrocolumns. This common architecture processes patterns of information within and among different regions of sensory, motor, associative cortex, etc. Therefore, the premise of this approach is that this is a good model to describe and analyze evolution/propagation of ideas among defined populations.

Relevant to this study is that a spatial-temporal lattice-field short-time conditional multiplicative-noise (nonlinear in drifts and diffusions) multivariate Gaussian-Markovian probability distribution is developed faithful to neocortical function/physiology. Such probability distributions are a basic input into the approach used here. The SMNI model was the first physical application of a nonlinear multivariate calculus developed by other mathematical physicists in the late 1970s to define a statistical mechanics of multivariate nonlinear nonequilibrium systems (Graham, 1977; Langouche et al., 1982).

SMNI Tests on STM and EEG

SMNI builds from synaptic interactions to minicolumnar, macrocolumnar, and regional interactions in neocortex. Since 1981, a series of SMNI papers has been developed model columns and regions of neocortex, spanning mm to cm of tissue. Most of these papers have dealt explicitly with calculating properties of STM and scalp EEG in order to test the basic formulation of this approach (Ingber, 1983; Ingber, 1985; Ingber & Nunez, 1995).

The SMNI modeling of local mesocolumnar interactions (convergence and divergence between minicolumnar and macrocolumnar interactions) was tested on STM phenomena. The SMNI modeling of macrocolumnar interactions across regions was tested on EEG phenomena.

SMNI Description of STM

SMNI studies have detailed that maximal numbers of attractors lie within the physical firing space of both excitatory and inhibitory minicolumnar firings, consistent with experimentally observed capacities of auditory and visual STM, when a "centering" mechanism is enforced by shifting background noise in synaptic interactions, consistent with experimental observations under conditions of selective attention (Ingber, 1985; Ingber, 1994).

These calculations were further supported by high-resolution evolution of the short-time conditional-probability propagator using PATHINT (Ingber & Nunez, 1995). SMNI correctly calculated the stability and duration of STM, the primacy versus recency rule,

Figure 1. Illustrated are three biophysical scales of neocortical interactions: (a)-(a*)-(a') microscopic neurons; (b)-(b') mesocolumnar domains; (c)-(c') macroscopic regions (Ingber, 1983). SMNI has developed appropriate conditional probability distributions at each level, aggregating up from the smallest levels of interactions. In (a*) synaptic inter-neuronal interactions, averaged over by mesocolumns, are phenomenologically described by the mean and variance of a distribution Ψ. Similarly, in (a) intraneuronal transmissions are phenomenologically described by the mean and variance of Γ. Mesocolumnar averaged excitatory (E) and inhibitory (I) neuronal firings M are represented in (a'). In (b) the vertical organization of minicolumns is sketched together with their horizontal stratification, yielding a physiological entity, the mesocolumn. In (b') the overlap of interacting mesocolumns at locations r and r' from times t and t + τ is sketched. In (c) macroscopic regions of neocortex are depicted as arising from many mesocolumnar domains. (c') sketches how regions may be coupled by long-ranged interactions.
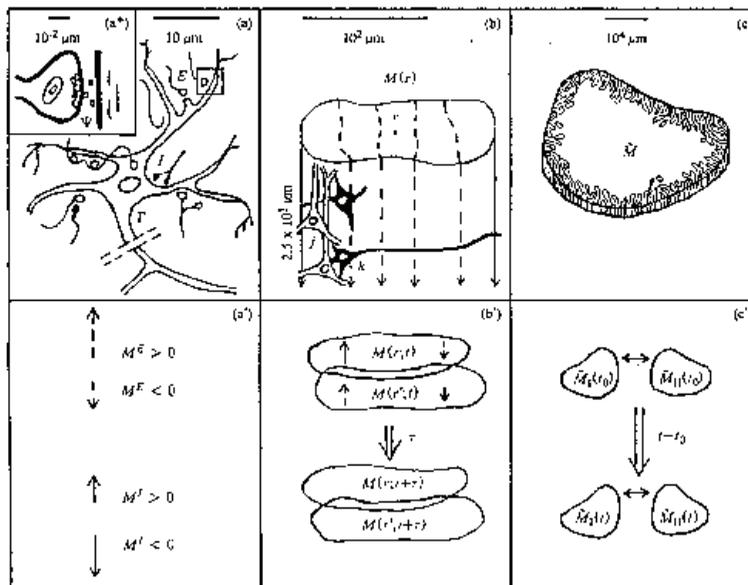
random access to memories within tenths of a second as observed, and the observed 7±2 capacity rule of auditory memory and the observed 4±2 capacity rule of visual memory.

SMNI also calculates how STM patterns (e.g., from a given region or even aggregated from multiple regions) may be encoded by dynamic modification of synaptic parameters (within experimentally observed ranges) into long-term memory patterns (LTM) (Ingber, 1983).

SMNI Description of EEG

Using the power of this formal structure, sets of EEG and evoked potential data from a separate NIH study, collected to investigate genetic predispositions to alcoholism, were fitted to an SMNI model on a lattice of regional electrodes to extract brain "signatures" of STM (Ingber, 1997). Each electrode site was represented by an SMNI distribution of independent stochastic macrocolumnar-scaled firing variables, interconnected by long-ranged circuitry with delays appropriate to long-fiber communication in neocortex. The global optimization algorithm ASA was used to perform maximum likelihood fits of Lagrangians defined by path integrals of multivariate conditional probabilities. Canonical momenta indicators (CMI) were thereby derived for individual's EEG data. The CMI give better signal recognition than the raw data, and were used to advantage as correlates of behavioral states. In-sample data was used for training (Ingber, 1997), and out-of-sample data was used for testing these fits. The architecture of ISM is modeled using scales similar to those used for local STM and global EEG connectivity.

Generic Mesoscopic Neural Networks

SMNI was applied to a parallelized generic mesoscopic neural networks (MNN) (Ingber, 1992), adding computational power to a similar paradigm proposed for target recognition.

"Learning" takes place by presenting the MNN with data, and parametrizing the data in terms of the firings, or multivariate firings. The "weights," or coefficients of functions of firings appearing in the drifts and diffusions, are fit to incoming data, considering the joint "effective" Lagrangian (including the logarithm of the prefactor in the probability distribution) as a dynamic

Figure 2. Scales of interactions among minicolumns are represented, within macrocolumns, across macrocolumns, and across regions of macrocolumns
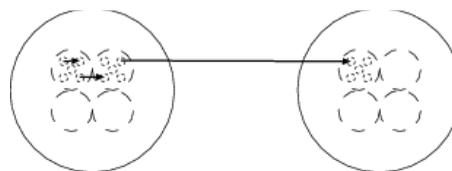
cost function. This program of fitting coefficients in Lagrangian uses methods of ASA. "Prediction" takes advantage of a mathematically equivalent representation of the Lagrangian path-integral algorithm, i.e., a set of coupled Langevin rate-equations. A coarse deterministic estimate to "predict" the evolution can be applied using the most probable path, but PATHINT has been used. PATHINT, even when parallelized, typically can be too slow for "predicting" evolution of these systems. However, PATHTREE is much faster.

Architecture for Selected ISM Model

The primary objective is to deliver a computer model that contains the following features: (1) A multivariable space will be defined to accommodate populations. (2) A cost function over the population variables in (1) will be defined to explicitly define a pattern that can be identified as an Idea. A very important issue is for this project is to develop cost functions, not only how to fit or process them. (3) Subsets of the population will be used to fit parameters — e.g, coefficients of variables, connectivities to patterns, etc. — to an Idea, using the cost function in (2). (4) Connectivity of the population in (3) will be made to the rest of the population. Investigations will be made to determine what endogenous connectivity is required to stop or promote the propagation of the Idea into other regions of the population. (5) External forces, e.g., acting only on specific regions of the population, will be introduced, to determine how these exogenous forces may stop or promote the propagation of an Idea.

Application of SMNI Model

The approach is to develop subsets of Ideas/macrocolumnar activity of multivariate stochastic descriptions of

defined populations (of a reasonable but small population samples, e.g., of 100-1000), with macrocolumns defined by their local parameters within specific regions (larger samples of populations) and with parameterized long-ranged inter-regional and external connectivities. Parameters of a given subset of macrocolumns will be fit using ASA to patterns representing Ideas, akin to acquiring hard-wired long-term (LTM) patterns. Parameters of external and inter-regional interactions will be determined that promote or inhibit the spread of these Ideas, by determining the degree of fits and overlaps of probability distributions relative to the seeded macrocolumns.

That is, the same Ideas/patterns may be represented in other than the seeded macrocolumns by local confluence of macrocolumnar and long-ranged firings, akin to STM, or by different hard-wired parameter LTM sets that can support the same local firings in other regions (possible in nonlinear systems). SMNI also calculates how STM can be dynamically encoded into LTM (Ingber, 1983).

Small populations in regions will be sampled to determine if the propagated Idea(s) exists in its pattern space where it did exist prior to its interactions with the seeded population. SMNI derives nonlinear functions as arguments of probability distributions, leading to multiple STM, e.g., 7 ± 2 for auditory memory capacity. Some investigation will be made into nonlinear functional forms other than those derived for SMNI, e.g., to have capacities of tens or hundreds of patterns for ISM.

Application of TRD Analysis

This approach includes application of methods of portfolio risk analysis to such statistical systems, correcting two kinds of errors committed in multivariate risk analyses: (E1) Although the distributions of variables being considered are not Gaussian (or not tested to see how close they are to Gaussian), standard statistical calculations appropriate only to Gaussian distributions are employed. (E2) Either correlations among the variables are ignored, or the mistakes committed in (E1) — incorrectly assuming variables are Gaussian — are compounded by calculating correlations as if all variables were Gaussian.

It should be understood that any sampling algorithm processing a huge number of states can find many multiple optima. ASA's MULTI_MIN OPTIONS are used to save multiple optima during sampling. Some algorithms might label these states as "mutations" of optimal states. It is important to be able to include them in final decisions, e.g., to apply additional metrics of performance specific to applications. Experience with risk-managing portfolios shows that all criteria are not best considered by lumping them all into one cost function, but rather good judgment should be applied to multiple stages of pre-processing and post-processing when performing such sampling, e.g., adding additional metrics of performance.

FUTURE TRENDS

Given financial and political motivations to merge information discussed in the Introduction, it is inevitable that many AI algorithms will be developed, and many current AI algorithms will be enhanced, to address these issues.

CONCLUSION

It seems appropriate to base an approach for propagation of generic ideas on the only system so far demonstrated to develop and nurture ideas, i.e., the neocortical brain. A statistical mechanical model of neocortical interactions, developed by the author and tested successfully in describing short-term memory and EEG indicators, Ideas by Statistical Mechanics (ISM) (Ingber, 2006) is the proposed model. ISM develops subsets of macrocolumnar activity of multivariate stochastic descriptions of defined populations, with macrocolumns defined by their local parameters within specific regions and with parameterized endogenous inter-regional and exogenous external connectivities. Tools of financial risk management, developed to process correlated multivariate systems with differing non-Gaussian distributions using modern copula analysis, importance-sampled using ASA, will enable bona fide correlations and uncertainties of success and failure to be calculated.

A

REFERENCES

Aletti, G., Naldi, G. & Toscani, G. (2006) First-order continuous models of opinion formation.

Report. U Milano. [Url http://lanl.arxiv.org/abs/cond-mat/0605092]

Allen, J. (2004) Commander's automated decision support tools. Report. DARPA. [URL http://www.darpa.mil/ato/solicit/IBC/allen.ppt]

Blanco, C. (2005) Financial Risk Management: Beyond Normality, Volatility and Correlations.

Financial Economics Network, Waltham, MA. [URL http://www.fenews.com/fen46/front-sr/blanco/blanco.html]

Borghesi, C. & Galam, S. (2006) Chaotic, staggered and polarized dynamics in opinion forming: the contrarian effect. Report. Service de Physique de l'Etat Condens. [Url http://lanl.arxiv.org/abs/physics/0605150]

Graham, R. (1977) Covariant formulation of non-equilibrium statistical thermodynamics. Zeitschrift fu¨r Physik. B26, 397-405.

Ingber, L. (1983) Statistical mechanics of neocortical interactions. Dynamics of synaptic modification. Physical Review A. 28, 395-416. [URL http://www.ingber.com/smni83_dynamics.pdf]

Ingber, L. (1985) Statistical mechanics of neocortical interactions: Stability and duration of the 7+-2 rule of short-term-memory capacity. Physical Review A. 31, 1183-1186. [URL http://www.ingber.com/smni85_stm.pdf]

Ingber, L. (1989) Very fast simulated re-annealing. Mathematical Computer Modelling. 12(8), 967-973. [URL http://www.ingber.com/asa89_vfsr.pdf]

Ingber, L. (1992) Generic mesoscopic neural networks based on statistical mechanics of neocortical interactions. Physical Review A. 45(4), R2183-R2186. [URL http://www.ingber.com/smni92_mnn.pdf]

Ingber, L. (1993) Adaptive Simulated Annealing (ASA). Global optimization C-code. Caltech Alumni Association. [URL http://www.ingber.com/#ASA-CODE]

Ingber, L. (1994) Statistical mechanics of neocortical interactions: Path-integral evolution of short-term memory. Physical Review E. 49(5B), 4652-4664. [URL http://www.ingber.com/smni94_stm.pdf]

Ingber, L. (1995) Statistical mechanics of multiple scales of neocortical interactions, In: Neocortical Dynamics and Human EEG Rhythms, ed. P.L. Nunez. Oxford University

Press, 628-681. [ISBN 0-19-505728-7. URL http://www.ingber.com/smni95_scales.pdf]

Ingber, L. (1997) Statistical mechanics of neocortical interactions: Applications of canonical momenta indicators to electroencephalography. Physical Review E. 55(4), 4578-4593. [URL http://www.ingber.com/smni97_cmi.pdf]

Ingber, L. (2006) Ideas by statistical mechanics (ISM). Report 2006:ISM. Lester Ingber Research. [URL http://www.ingber.com/smni06_ism.pdf]

Ingber, L. & Nunez, P.L. (1995) Statistical mechanics of neocortical interactions: High resolution path-integral calculation of short-term memory. Physical Review E. 51(5), 5074-5083. [URL http://www.ingber.com/smni95_stm.pdf]

Ingber, L. & Rosen, B. (1992) Genetic algorithms and very fast simulated reannealing: A comparison. Mathematical Computer Modelling. 16(11), 87-100. [URL http://www.ingber.com/asa92_saga.pdf]

Langouche, F., Roekaerts, D. & Tirapegui, E. (1982) Functional Integration and Semiclassical Expansions. Reidel, Dordrecht, The Netherlands.

Sen, P. (2006) Complexities of social networks: A physicist's perspective. Report. U Calcutta. [Url http://lanl.arxiv.org/abs/physics/0605072]

Situngkir, H. (2004) On selfish memes: Culture as complex adaptive system. Journal Social Complexity. 2(1), 20-32. [URL http://cogprints.org/3471/]

Thurtle, P.S. (2006) "The G Files": Linking "The Selfish Gene" And "The Thinking Reed".

Stanford Presidential Lectures and Symposia in the Humanities and Arts. Standford U. [URL http://prelectur.stanford.edu/lecturers/gould/commentary/thurtle.html]

KEY TERMS

Copula Analysis: This transforms non-Gaussian probability distributions to a common appropriate space (usually a Gaussian space) where it makes sense to calculate correlations as second moments.

DIME: Represents diplomatic, information, military, and economic aspects of information that must be merged into coherent pattern.

Global Optimization: Refers to a collection of algorithms used to statistically sample a space of parameters or variables to optimize a system, but also often used to sample a huge space for information. There are many variants, including simulated annealing, genetic algorithms, ant colony optimization, hill-climbing, etc.

ISM: An anacronym for Ideas by Statistical Mechanics in the context of the noun defined as: A belief (or system of beliefs) accepted as authoritative by some group or school. A doctrine or theory; especially, a wild or visionary theory. A distinctive doctrine, theory, system, or practice.

Meme: Alludes to a technology originally defined to explain social evolution, which has been refined to mean a gene-like analytic tool to study cultural evolution.

Memory: This may have many forms and mechanisms. Here, two major processes of neocortical memory are used for AI technologies, short-term memory (STM) and long-term memory (LTM).

Simulated Annealing (SA): A class of algorithms for sampling a huge space, which has a mathematical proof of convergence to global optimal minima. Most SA algorithms applied to most systems do not fully take advantage of this proof, but the proof often is useful to give confidence that the system will avoid getting stuck for a long time in local optimal regions.

Statistical Mechanics: A branch of mathematical physics dealing with systems with a large number of states. Applications of nonequilibrium nonlinear statistical mechanics are now common in many fields, ranging from physical and biological sciences, to finance, to computer science, etc.

AI Methods for Analyzing Microarray Data

Amira Djebbari
National Research Council Canada, Canada

Aedín C. Culhane
Harvard School of Public Health, USA

Alice J. Armstrong
The George Washington University, USA

John Quackenbush
Harvard School of Public Health, USA

INTRODUCTION

Biological systems can be viewed as information management systems, with a basic instruction set stored in each cell's DNA as "genes." For most genes, their information is enabled when they are transcribed into RNA which is subsequently translated into the proteins that form much of a cell's machinery. Although details of the process for individual genes are known, more complex interactions between elements are yet to be discovered. What we do know is that diseases can result if there are changes in the genes themselves, in the proteins they encode, or if RNAs or proteins are made at the wrong time or in the wrong quantities.

Recent advances in biotechnology led to the development of DNA microarrays, which quantitatively measure the expression of thousands of genes simultaneously and provide a snapshot of a cell's response to a particular condition. Finding patterns of gene expression that provide insight into biological endpoints offers great opportunities for revolutionizing diagnostic and prognostic medicine and providing mechanistic insight in data-driven research in the life sciences, an area with a great need for advances, given the urgency associated with diseases. However, microarray data analysis presents a number of challenges, from noisy data to the curse of dimensionality (large number of features, small number of instances) to problems with no clear solutions (*e.g.* real world mappings of genes to traits or diseases that are not yet known).

Finding patterns of gene expression in microarray data poses problems of class discovery, comparison, prediction, and network analysis which are often approached with AI methods. Many of these methods have been successfully applied to microarray data analysis in a variety of applications ranging from clustering of yeast gene expression patterns (Eisen *et al.*, 1998) to classification of different types of leukemia (Golub *et al.*, 1999). Unsupervised learning methods (*e.g.* hierarchical clustering) explore clusters in data and have been used for class discovery of distinct forms of diffuse large B-cell lymphoma (Alizadeh *et al.*, 2000). Supervised learning methods (*e.g.* artificial neural networks) utilize a previously determined mapping between biological samples and classes (*i.e.* labels) to generate models for class prediction. A k-nearest neighbor (k-NN) approach was used to train a gene expression classifier of different forms of brain tumors and its predictions were able to distinguish biopsy samples with different prognosis suggesting that microarray profiles can predict clinical outcome and direct treatment (Nutt *et al.*, 2003). Bayesian networks constructed from microarray data hold promise for elucidating the underlying biological mechanisms of disease (Friedman *et al.*, 2000).

BACKGROUND

Cells dynamically respond to their environment by changing the set and concentrations of active genes by altering the associated RNA expression. Thus "gene expression" is one of the main determinants of a cell's state, or phenotype. For example, we can investigate the differences between a normal cell and a cancer cell by examining their relative gene expression profiles.

Microarrays quantify gene expression levels in various conditions (such as disease *vs.* normal) or across time points. For n genes and m instances (biological

Table 1. Some public online repositories of microarray data

Name of the repository	URL
ArrayExpress at the European Bioinformatics Institute	http://www.ebi.ac.uk/arrayexpress/
Gene Expression Omnibus at the National Institutes of Health	http://www.ncbi.nlm.nih.gov/geo/
Stanford microarray database	http://smd.stanford.edu/
Oncomine	http://www.oncomine.org/main/index.jsp

samples), microarray measurements are stored in an *n* by *m* matrix where each row is a gene, each column is a sample and each element in the matrix is the expression level of a gene in a biological sample, where samples are instances and genes are features describing those instances. Microarray data is available through many public online repositories (Table 1). In addition, the Kent-Ridge repository (http://sdmc.i2r.a-star.edu.sg/rp/) contains pre-formatted data ready to use with the well-known machine learning tool Weka (Witten & Frank, 2000).

Microarray data presents some unique challenges for AI such as a severe case of the curse of dimensionality due to the scarcity of biological samples (instances). Microarray studies typically measure tens of thousands of genes in only tens of samples. This low case to variable ratio increases the risk of detecting spurious relationships. This problem is exacerbated because microarray data contains multiple sources of within-class variability, both technical and biological. The high levels of variance and low sample size make feature selection difficult. Testing thousands of genes creates a multiple testing problem, which can result in under-estimating the number of false positives. Given data with these limitations, constructing models becomes under-determined and therefore prone to over-fitting.

From biology, it is also clear that genes do not act independently. Genes interact in the form of pathways or gene regulatory networks. For this reason, we need models that can be interpreted in the context of pathways. Researchers have successfully applied AI methods to microarray data preprocessing, clustering, feature selection, classification, and network analysis.

MINING MICROARRAY DATA: CURRENT TECHNIQUES, CHALLENGES AND OPPORTUNITIES FOR AI

Data Preprocessing

After obtaining microarray data, normalization is performed to account for systematic measurement biases and to facilitate between-sample comparisons (Quackenbush, 2002). Microarray data may contain missing values that may be replaced by mean replacement or k-NN imputation (Troyanskaya *et al.*, 2001).

Feature Selection

The goal of feature selection is to find genes (features) that best distinguish groups of instances (*e.g.* disease *vs.* normal) to reduce the dimensionality of the dataset. Several statistical methods including t-test, significance analysis of microarrays (SAM) (Tusher *et al.*, 2001), and analysis of variance (ANOVA) have been applied to select features from microarray data.

In classification experiments, feature selection methods generally aim to identify relevant gene subsets to construct a classifier with good performance (Inza *et al.*, 2004). Features are considered to be relevant when they can affect the class; the strongly relevant are indispensable to prediction and the weakly relevant may only sometimes contribute to prediction.

Filter methods evaluate feature subsets regardless of the specific learning algorithm used. The statistical methods for feature selection discussed above as well as rankers like information gain rankers are filters for the features to be included. These methods ignore the fact that there may be redundant features (features that are highly correlated with each other and as such one can be used to replace the other) and so do not seek to find a set of features which could perform similarly

with fewer variables while retaining the same predictive power (Guyon & Elisseeff, 2003). For this reason multivariate methods are more appropriate.

As an alternative, wrappers consider the learning algorithm as a black-box and use prediction accuracy to evaluate feature subsets (Kohavi & John, 1997). Wrappers are more direct than filter methods but depend on the particular learning algorithm used. The computational complexity associated with wrappers is prohibitive due to curse of dimensionality, so typically filters are used with forward selection (starting with an empty set and adding features one by one) instead of backward elimination (starting with all features and removing them one by one). Dimension reduction approaches are also used for multivariate feature selection.

Dimension Reduction Approaches

Principal component analysis (PCA) is widely used for dimension reduction in machine learning (Wall *et al.*, 2003). The idea behind PCA is quite intuitive: correlated objects can be combined to reduce data "dimensionality". Relationships between gene expression profiles in a data matrix can be expressed as a linear combination such that colinear variables are regressed onto a new set of coordinates. PCA, its underlying method Single Value Decomposition (SVD), related approaches such as correspondence analysis (COA), and multidimensional scaling (MDS) have been applied to microarray data and are reviewed by Brazma & Culhane (2005). Studies have reported that COA or other dual scaling dimension reduction approaches such as spectral map analysis may be more appropriate than PCA for decomposition of microarray data (Wouters *et al.*, 2003).

While PCA considers the variance of the whole dataset, clustering approaches examine the pairwise distance between instances or features. Therefore, these methods are complementary and are often both used in exploratory data analysis. However, difficulties in interpreting the results in terms of discrete genes limit the application of these methods.

Clustering

What we see as one disease is often a collection of disease subtypes. Class discovery aims to discover these subtypes by finding groups of instances with similar expression patterns. Hierarchical clustering is an agglomerative method which starts with a singleton

and groups similar data points using some distance measure such that two data points that are most similar are grouped together in a cluster by making them children of a parent node in the tree. This process is repeated in a bottom-up fashion until all data points belong to a single cluster (corresponding to the root of the tree).

Hierarchical and other clustering approaches, including K-means, have been applied to microarray data (Causton *et al.*, 2003). Hierarchical clustering was applied to study gene expression in samples from patients with diffuse large B-cell lymphoma (DLBCL) resulting in the discovery of two subtypes of the disease. These groups were found by analyzing microarray data from biopsy samples of patients who had not been previously treated. These patients continued to be studied after chemotherapy, and researchers found that the two newly discovered disease subtypes had different survival rates, confirming the hypothesis that the subtypes had significantly different pathologies (Alizadeh *et al.*, 2000).

While clustering simply groups the given data based on pair-wise distances, when information is known *a priori* about some or all of the data *i.e.* labels, a supervised approach can be used to obtain a classifier that can predict the label of new instances.

Classification (Supervised Learning)

The large dimensionality of microarray data means that all classification methods are susceptible to over-fitting. Several supervised approaches have been applied to microarray data including Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and k-NNs among others (Hastie *et al.*, 2001).

A very challenging and clinically relevant problem is the accurate diagnosis of the primary origin of metastatic tumors. Bloom *et al.* (2004) applied ANNs to the microarray data of 21 tumor types with 88% accuracy to predict the primary site of origin of metastatic cancers with unknown origin. A classification of 84% was obtained on an independent test set with important implications for diagnosing cancer origin and directing therapy.

In a comparison of different SVM approaches, multicategory SVMs were reported to outperform other popular machine learning algorithms such as k-NNs and ANNs (Statnikov *et al.*, 2005) when applied to 11 publicly available microarray datasets related to cancer.

It is worth noting that feature selection can significantly improve classification performance.

Cross-Validation

Cross-validation (CV) is appropriate in microarray studies which are often limited by the number of instances (*e.g.* patient samples). In k-fold CV, the training set is divided into *k* subsets of equal size. In each iteration *k*-1 subsets are used for training and one subset is used for testing. This process is repeated *k* times and the mean accuracy is reported. Unfortunately, some published studies have applied CV only partially, by applying CV on the creation of the prediction rule while excluding feature selection. This introduces a bias in the estimated error rates and over-estimates the classification accuracy (Simon *et al.*, 2003). As a consequence, results from many studies are controversial due to methodological flaws (Dupuy & Simon, 2007). Therefore, models must be evaluated carefully to prevent selection bias (Ambroise & McLachlan, 2002). Nested CV is recommended, with an inner CV loop to perform the tuning of the parameters and an outer CV to compute an estimate of the error (Varma & Simon, 2006).

Several studies which have examined similar biological problems have reported poor overlap in gene expression signatures. Brenton *et al.* (2005) compared two gene lists predictive of breast cancer prognosis and found only 3 genes in common. Even though the intersection of specific gene lists is poor, the highly correlated nature of microarray data means that many gene lists may have similar prediction accuracy (Ein-Dor *et al.*, 2004). Gene signatures identified from different breast cancer studies with few genes in common were shown to have comparable success in predicting patient survival (Buyse *et al.*, 2006).

Commonly used supervised learning algorithms yield black box models prompting the need for interpretable models that provide insights about the underlying biological mechanism that produced the data.

Network Analysis

Bayesian networks (BNs), derived from an alliance between graph theory and probability theory, can capture dependencies among many variables (Pearl, 1988, Heckerman, 1996).

Friedman *et al.* (2000) introduced a multinomial model framework for BNs to reverse-engineer networks and showed that this method differs from clustering in that it can discover gene interactions other than correlation when applied to yeast gene expression data. Spirtes *et al.* (2002) highlight some of the difficulties of applying this approach to microarray data. Nevertheless, many extensions of this research direction have been explored. Correlation is not necessarily a good predictor of interactions, and weak interactions are essential to understand disease progression. Identifying the biologically meaningful interactions from the spurious ones is challenging, and BNs are particularly well-suited for modeling stochastic biological processes.

The exponential growth of data produced by microarray technology as well as other high-throughput data (*e.g.* protein-protein interactions) call for novel AI approaches as the paradigm shifts from a reductionist to a mechanistic systems view in the life sciences.

FUTURE TRENDS

Uncovering the underlying biological mechanisms that generate these data is harder than prediction and has the potential to have far reaching implications for understanding disease etiologies. Time series analysis (Bar-Joseph, 2004) is a first step to understanding the dynamics of gene regulation, but, eventually, we need to use the technology not only to observe gene expression data but also to direct intervention experiments (Pe'er *et al.*, 2001, Yoo *et al.*, 2002) and develop methods to investigate the fundamental problem of distinguishing correlation from causation.

CONCLUSION

We have reviewed AI methods for pre-processing, clustering, feature selection, classification and mechanistic analysis of microarray data. The clusters, gene lists, molecular fingerprints and network hypotheses produced by these approaches have already shown impact; from discovering new disease subtypes and biological markers, predicting clinical outcome for directing treatment as well as unraveling gene networks. From the AI perspective, this field offers challenging problems and may have a tremendous impact on biology and medicine.

REFERENCES

Alizadeh A.A., Eisen M.B., Davis R.E., Ma C., Lossos I.S., Rosenwald A., *et al.* (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature, 403*(6769), 503-11.

Ambroise C., & McLachlan G.J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences,* 99(10), 6562-6.

Bar-Joseph Z. (2004). Analyzing time series gene expression data. *Bioinformatics, 20*(16), 2493-503.

Bloom G., Yang I.V., Boulware D., Kwong K.Y., Coppola D., Eschrich S., *et al.* (2004). Multi-platform, multi-site, microarray-based human tumor classification. *American Journal of Pathology, 164*(1), 9-16.

Brenton J.D., Carey L.A., Ahmed A.A., & Caldas C. (2005). Molecular classification and molecular forecasting of breast cancer: ready for clinical application? *Journal of Clinical Oncology, 23*(29), 7350-60.

Brazma A., & Culhane AC. (2005). Algorithms for gene expression analysis. In Jorde LB., Little PFR, Dunn MJ., Subramaniam S. (Eds.) *Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics.,* (3148 -3159) London: John Wiley & Sons.

Buyse, M., Loi S., Van't Veer L., Viale G., Delorenzi M., Glas A.M., *et al.* (2006). Validation and clinical utility of a 70-gene prognostic signature for women with node-negative breast cancer. *Journal of the National Cancer Institute, 98*, 1183-92.

Causton H.C., Quackenbush J., & Brazma A. (2003) Microarray Gene Expression Data Analysis: A Beginner's Guide. Oxford: Blackwell Science Limited.

Dupuy A., & Simon RM. (2007). Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. *Journal of the National Cancer Institute, 99*(2), 147-57.

Ein-Dor L., Kela I., Getz G., Givol D., & Domany E. (2004). Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics, 21*(2), 171-8.

Eisen M.B., Spellman P.T., Brown P.O., & Botstein D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences, 95*, 14863-14868.

Friedman N., Linial M., Nachman I., & Pe'er D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology, 7*(3-4), 601-20.

Golub T. R., Slonim D. K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J. P., *et al.* (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science, 286* (5439), 531.

Guyon, I., & Elisseff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157-1182.

Hastie T., Tibshirani R., & Friedman J. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer Series in Statistics.

Heckerman D. (1996). A Tutorial on Learning with Bayesian Networks. Technical Report MSR-TR-95-06. Microsoft Research.

Inza I., Larrañaga P., Blanco R., & Cerrolaza A.J. (2004). Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine, special issue in "Data mining in genomics and proteomics", 31*(2), 91-103.

Kohavi R., & John G.H. (1997). Wrappers for feature subset selection, *Artificial Intelligence, 97*(1-2), 273-324.

Nutt C.L., Mani D.R., Betensky R.A., Tamayo P., Cairncross J.G., Ladd C., *et al.* (2003). Gene Expression-based Classification of Malignant Gliomas Correlates Better with Survival than Histological Classification. *Cancer Research, 63*, 1602-1607.

Pe'er D, Regev A, Elidan G, & Friedman N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics, 17 S1*, S215-24.

Pearl J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, San Mateo: Morgan Kaufmann Publishers.

Quackenbush J. (2002). Microarray data normalization and transformation, *Nature Genetics, 32*, 496–501.

Quackenbush J. (2006). Microarray Analysis and Tumor Classification. *The New England Journal of Medicine, 354*(23), 2463-72.

A

Simon R., Radmacher M.D., Dobbin K., & McShane L.M. (2003). Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *Journal of the National Cancer Institute, 95*(1), 14-8.

Spirtes, P., Glymour, C., Scheines, R. Kauffman, S., Aimale, V., & Wimberly, F. (2001). Constructing Bayesian Network Models of Gene Expression Networks from Microarray Data. *Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology.*

Statnikov A., Aliferis C.F., Tsamardinos I., Hardin D., & Levy S. (2005). A comprehensive evaluation of multicategory classification methodsfor microarray gene expression cancer diagnosis. *Bioinformatics, 21*(5), 631-643

Troyanskaya O., Cantor M., Sherlock G., Brown P., Hastie T., Tibshirani R., *et al.* (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics, 17*(6), 520-5.

Tusher V.G., Tibshirani R., & Chu G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences, 98*(9), 5116-5121.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics, 7*, 91

Witten, I. H. & Frank, E. (2000). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers Inc.

Wall, M., Rechtsteiner, A., & Rocha, L. (2003). Singular value decomposition and principal component analysis. In D.P. Berrar, W. Dubitzky, M. Granzow (Eds.) *A Practical Approach to Microarray Data Analysis.* (91-109). Norwell: Kluwer.

Wouters, L., Gohlmann, H.W., Bijnens, L., Kass, S.U., Molenberghs, G., & Lewi, P.J. (2003). Graphical exploration of gene expression data: a comparative study of three multivariate methods. *Biometrics, 59*, 1131-1139

Yoo C., Thorsson V., & Cooper G.F. (2002). Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. *Biocomputing: Proceedings of the Pacific Symposium, 7*, 498-509

KEY TERMS

Curse of Dimensionality: A situation where the number of features (genes) is much larger than the number of instances (biological samples) which is known in statistics as $p \gg n$ problem.

Feature Selection: A problem of finding a subset (or subsets) of features so as to improve the performance of learning algorithms.

Microarray: A microarray is an experimental assay which measures the abundances of mRNA (intermediary between DNA and proteins) corresponding to gene expression levels in biological samples.

Multiple testing problem: A problem that occurs when a large number of hypotheses are tested simultaneously using a user-defined α cut off p-value which may lead to rejecting a non-negligible number of null hypotheses by chance.

Over-Fitting: A situation where a model learns spurious relationships and as a result can predict training data labels but not generalize to predict future data.

Supervised Learning: A learning algorithm that is given a training set consisting of feature vectors associated with class labels and whose goal is to learn a classifier that can predict the class labels of future instances.

Unsupervised Learning: A learning algorithm that tries to identify clusters based on similarity between features or between instances or both but without taking into account any prior knowledge.

An AI Walk from Pharmacokinetics to Marketing

José D. Martín-Guerrero
University of Valencia, Spain

Emilio Soria-Olivas
University of Valencia, Spain

Paulo J.G. Lisboa
Liverpool John Moores University, UK

Antonio J. Serrano-López
University of Valencia, Spain

INTRODUCTION

This work is intended for providing a review of real-life practical applications of Artificial Intelligence (AI) methods. We focus on the use of Machine Learning (ML) methods applied to rather real problems than synthetic problems with standard and controlled environment. In particular, we will describe the following problems in next sections:

- Optimization of Erythropoietin (EPO) dosages in anaemic patients undergoing Chronic Renal Failure (CRF).
- Optimization of a recommender system for citizen web portal users.
- Optimization of a marketing campaign.

The choice of these problems is due to their relevance and their heterogeneity. This heterogeneity shows the capabilities and versatility of ML methods to solve real-life problems in very different fields of knowledge. The following methods will be mentioned during this work:

- Artificial Neural Networks (ANNs): Multilayer Perceptron (MLP), Finite Impulse Response (FIR) Neural Network, Elman Network, Self-Oganizing Maps (SOMs) and Adaptive Resonance Theory (ART).
- Other clustering algorithms: K-Means, Expectation-Maximization (EM) algorithm, Fuzzy C-Means (FCM), Hierarchical Clustering Algorithms (HCA).
- Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH).
- Support Vector Regression (SVR).
- Collaborative filtering techniques.
- Reinforcement Learning (RL) methods.

BACKGROUND

The aim of this communication is to emphasize the capabilities of ML methods to deliver practical and effective solutions in difficult real-world applications. In order to make the work easy to read we focus on each of the three separate domains, namely, Pharmacokinetics (PK), Web Recommender Systems and Marketing.

Pharmacokinetics

Clinical decision-making support systems have used Artificial Intelligence (AI) methods since the end of the fifties. Nevertheless, it was only during the nineties that decision support systems were routinely used in clinical practice on a significant scale. In particular, ANNs have been widely used in medical applications the last two decades (Lisboa, 2002). One of the first relevant studies involving ANNs and Therapeutic Drug Monitoring was (Gray, Ash, Jacobi, & Michel, 1991). In this work, an ANN-based drug interaction warning system was developed with a computerized real-time entry medical records system. A reference work in this field is found in (Brier, Zurada, & Aronoff, 1995), in which the capabilities of ANNs and NONMEN are benchmarked.

Focusing on problems that are closer to the real-life application that will be described in next section, there are also a number of recent works involving the use of ML for drug delivery in kidney disease. For instance, a comparison of renal-related adverse drug reactions between rofecoxib and celecoxib, based on the WHO/Uppsala Monitoring Centre safety database, was carried out by (Zhao, Reynolds, Lejkowith, Whelton, & Arellano, 2001). Disproportionality in the association between a particular drug and renal-related adverse drug reactions was evaluated using a Bayesian confidence propagation neural network method. A study of prediction of cyclosporine dosage in patients after kidney transplantation using neural networks and kernel-based methods was carried out in (Camps et al., 2003). In (Gaweda, Jacobs, Brier, & Zurada, 2003), a pharmacodynamic population analysis in CRF patients using ANNs was performed. Such models allow for adjusting the dosing regime. Finally, in (Martín et al., 2003) , the use of neural networks was proposed for the optimization of EPO dosage in patients undergoing anaemia connected with CRF.

Web Recommender Systems

Recommender systems are widely used in web sites including Google. The main goal of these systems is to recommend objects which a user might be interested in. Two main approaches have been used: content-based and collaborative filtering (Zukerman & Albrecht, 2001), although other kinds of techniques have also been proposed (Burke, 2002).

Collaborative recommenders aggregate ratings of recommendations of objects, find user similarities based on their ratings, and finally provide new recommendations based on inter-user comparisons. Some of the most relevant systems using this technique are GroupLens/NetPerceptions and Recommender. The main advantage of collaborative techniques is that they are independent from any machine-readable representation of the objects, and that they work well for complex objects where subjective judgements are responsible for much of the variation in preferences.

Content-based learning is used when a user's past behaviour is a reliable indicator of his/her future behaviour. It is particularly suitable for situations in which users tend to exhibit idiosyncratic behaviour. However, this approach requires a system to collect relatively large amounts of data from each user in order to enable the formulation of a statistical model. Examples of systems of this kind are text recommendation systems like the newsgroup filtering system, NewsWeeder, which uses words from its texts as features.

Marketing

The latest marketing trends are more concerned about maintaining current customers and optimizing their behaviour than getting new ones. For this reason, relational marketing focuses on what a company must do to achieve this objective. The relationships between a company and its costumers follow a sequence of action-response system, where the customers can modify their behaviour in accordance with the marketing actions developed by the company.

The development of a good and individualized policy is not easy because there are many variables to take into account. Applications of this kind can be viewed as a Markov chain problem, in which a company decides what action to take once the customer properties in the current state (time t), are known. Reinforcement Learning (RL) can be used to solve this task since previous applications have demonstrated its suitability in this area. In (Sun, 2003), RL was applied to analyse mailing by studying how an action in time t influences actions in following times. In (Abe et al., 2002) and (Pednault, Abe & Zadrozny., 2002), several RL algorithms were benchmarked in mailing problems. In (Abe, 2004), RL was used to optimize cross channel marketing.

AI CONTRIBUTIONS IN REAL-LIFE APPLICATIONS

Previous section showed a review of related work. In this section, we will focus on showing authors' experience in using AI to solve real-life problems. In order to show up the versatility of AI methods, we will focus on particular applications from three different fields of knowledge, the same that were reviewed in previous section.

Pharmacokinetics

Although we have also worked with other pharmacokinetic problems, in this work, we focus on maybe the most relevant problem, which is the

optimization of EPO dosages in patients within a haemodialysis program. Patients who suffer from CRF tend to suffer from an associated anaemia, as well. EPO is the treatment of choice for this kind of anaemia. The use of this drug has greatly reduced cardiovascular problems and the necessity of multiple transfusions. However, EPO is expensive, making the already costly CRF program even more so. Moreover, there are significant risks associated with EPO such as thrombo-embolisms and vascular problems, if Haemoglobin (Hb) levels are too high or they increase too fast. Consequently, optimizing dosage is critical to ensure adequate pharmacotherapy as well as a reasonable treatment cost.

Population models, widely used by Pharmacokinetics' researchers, are not suitable for this problem since the response to the treatment with EPO is highly dependent on the patient. The same dosages may have very different responses in different patients, most notably the so-called EPO-resistant patients, who do not respond to EPO treatment, even after receiving high dosages. Therefore, it is preferable to focus on an individualized treatment.

Our first approach to this problem was based on predicting the Hb level given a certain administered dose of EPO. Although the final goal is to individualize EPO doses, we did not predict EPO dose but Hb level. The reason is that EPO predictors would model physician's protocol whereas Hb predictors model body's response to the treatment, hence being a more "objective" approach. In particular, the following models were used: GARCH (Hamilton, 1994), MLP, FIR neural network, Elman's recurrent neural network and SVR (Haykin, 1999). Accurate prediction models were obtained, especially when using ANNs and SVR. Dynamic neural networks (i.e., FIR and recurrent) did not outperform notably the static MLP probably due to the short length of the time series (Martín et al., 2003). An easy-to-use software application was developed to be used by clinicians, in which after filling in patients' data and a certain EPO dose, the predicted Hb level for next month was shown.

Although prediction models were accurate, we realized that this prediction approach had a major flaw. Despite obtaining accurate models, we had not yet achieved a straightforward way to transfer the extracted knowledge to daily clinical practice, because clinicians had to "play" with different doses to analyse the best solution to attain a certain Hb level. It would be better to have an automatic model that suggests the actions to be made in order to attain the targeted range of Hb, rather than this "indirect" approach. This reflection made us research on new models, and we came up with the use of RL (Sutton & Barto, 1998). We are currently working on this topic but we have already achieved promising results, finding policies (sequence of actions) that appear to be better than those followed in the hospital, i.e., there are a higher number of patients within the desired target of Hb at the end of the treatment (Martín et al., 2006a).

Web Recommender Systems

A completely different application is described in this subsection, namely, the development of web recommender systems. The authors proposed a new approach to develop recommender systems based on collaborative filtering, but also including an analysis of the feasibility of the recommender by using a prediction stage (Martín et al., 2006b).

The very basic idea was to use clustering algorithms in order to find groups of similar users. The following clustering algorithms were taken into account: K-Means, FCM, HCA, EM algorithm, SOMs and ART. New users were assigned to one of the groups found by these clustering algorithms, and then they were recommended with web services that were usually accessed by other users of his/her same group, but had not yet been accessed by these new users (in order to maximize the usefulness of the approach). Using controlled data sets, the study concluded that ART and SOMs showed a very good behaviour with data sets of very different characteristics, whereas HCA and EM showed an acceptable behaviour provided that the dimensionality of the data set was not too high and the overlap was slight. Algorithms based on K-Means achieved the most limited success in the acceptance of offered recommendations.

Even though the use of RL was only slightly studied, it seems to be a suitable choice for this problem, since the internal dynamics of the problem is easily tackled by RL, and moreover the interference between the recommendation interface and the user can be minimized with an adequate definition of the rewards (Hernández, Gaudioso, & Boticario, 2004).

Marketing

The last application that will be mentioned in this communication is related to marketing. One way to increase the loyalty of customers is by offering them the opportunity to obtain some gifts as the result of their purchases from a certain company. The company can give *virtual credits* to anyone who buys certain articles, typically those that the company is interested in promoting. After a certain number of purchases, the customers can exchange their virtual credits for the gifts offered by the company. The problem is to establish the appropriate number of virtual credits for each promoted item. In accordance with the company policy, it is expected that the higher the credit assignment, the higher the amount of purchases. However, the company's profits are lower since the marketing campaign adds an extra cost to the company. The goal is to achieve a trade-off by establishing an optimal policy.

We proposed a RL approach to optimize this marketing campaign. This particular application, whose characteristics are described below, is much more difficult than the other RL approaches to marketing mentioned in the Background Section. This is basically because there are many more different actions that can be taken. The information used for the study corresponds to five months of the campaign, involving 1,264,862 transactions, 1,004 articles and 3,573 customers.

RL can deal with intrinsic dynamics, and besides, it has the attractive advantage that is able to maximize the so-called long-term reward. This is especially relevant in this application since the company is interested in maximizing the profits at the end of the campaign, and a customer who do not produce much profits in the first months of the campaign, may however make many profitable transactions in the future.

Our first results showed that profits using a policy based on RL instead of the policy followed by the company so far, could even double long-term profits at the end of the campaign (Gómez et al., 2005).

CONCLUSION AND FUTURE TRENDS

This paper has shown the capabilities and versatility of different AI methods to be applied to real-life problems, illustrated with three specific applications in different domains. Clearly, the methodology is generic and applies equally well to many other fields,

provided that the information contained in the data is sufficiently rich to require non-linear modelling and is capable of supporting a predictive performance that is of practical value.

As a next future trend, it should be emphasized that AI methods are increasingly popular for business applications in recent years, challenging classical business models.

In the particular case of RL, the commercial potential of this powerful methodology has been significantly underestimated, as it is applied almost exclusively to Robotics. We feel that it is a methodology still to be exploited in many real applications, as we have shown in this paper.

REFERENCES

Abe, N., Pednault, E., Wang, H., Zadrozny, B., Wei, F., & Apte, C. (2002). Empirical comparison of various reinforcement learning strategies for sequential targeted marketing. *Proceedings of the ICDM 2002*, 315-321.

Abe, N., Verma, N., Schroko, R. & Apte, C. (2004). Cross-channel optimized marketing by reinforcement learning. *Proceedings of the KDD 2004*, 767-772.

Brier, M. E., Zurada, J. M., & Aronoff, G. R. (1995). Neural network predicted peak and trough gentamicin concentrations. *Pharmaceutical Research*, 12 (3), 406-412.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12, 331-370.

Camps, G., Porta, B., Soria, E., Martín, J. D., Serrano, A. J., Pérez, J. J., & Jiménez, N. V. (2003). Prediction of cyclosporine dosage in patients after kidney transplantation using neural networks. *IEEE Transactions on Biomedical Engineering*, 50 (4), 442-448.

Gaweda, A. E., Jacobs, A. A., Brier, M. E., & Zurada, J. M. (2003). Pharmacodynamic population analysis in chronic renal failure using artificial neural networks – a comparative study. *Neural Networks*, 16 (5-6), 841-845.

Gómez, G., Martín, J. D., Soria, E., Palomares, A., Balaguer, E., Casariego, N.,, & Paglialunga, D. (2005). An approach based on reinforcement learning and

Seelf-Organizing Maps to design a marketing campaign. *Proceedings of the 2ⁿᵈ International Conference on Machine Intelligence ACIDCA-ICMI 2005,* 259-265.

Gray, D. L., Ash, S. R., Jacobi, J. & Michel, A. N. (1991). The training and use of an artificial neural network to monitor use of medication in treatment of complex patients. *Journal of Clinical Engineering,* 16 (4), 331-336.

Hamilton, J. D. (1994). *Time Series Analysis*, Princeton University Press, Princeton NJ, USA.

Haykin, S. (1999). *Neural Networks* (2nd ed.). Prentice Hall, Englewood Cliffs, NJ, USA.

Hernández, F., Gaudioso, E. & Boticario, J. G. (2004) A reinforcement approach to achieve unobstrusive and interactive recommendation systems for web-based communities. *Proceedings of Adaptive Hypermedia 2004*, 409-412.

Lisboa, P. J. G. (2002). A review of evidence of health benefit from artificial neural networks in medical intervention. *Neural Networks*, 15 (1), 11-39.

Martín, J. D., Soria, E., Camps, G., Serrano, A. J., Pérez, J. J., & Jiménez, N. V. (2003). Use of neural networks for dosage indidualisation of erythropoietin in patients with secondary anemia to chronic renal failure. *Computers in Biology and Medicine,* 33 (4), 361-373.

Martín, J. D., Soria, E., Chorro, V., Climente. M., & Jiménez, N. V. (2006a). Reinforcement Learning for anemia management in hemodialysis patients treated with erythropoietic stimulating factors. *Proceedings of the Workshop "Planning, Learning and Monitoring with uncertainty and dynamic worlds", European Conference on Artificial Intelligence 2006,* 19-24.

Martín, J. D., Palomares, A., Balaguer, E., Soria, E., Gómez, J., & Soriano, A. (2006b) Studying the feasibility of a recommender in a citizen web portal based on user modeling and clustering algorithms. *Expert Systems with Aplications*, 30 (2), 299-312.

Pednault, E., Abe, N., & Zadrozny, B. (2002). Sequential cost-sensitive decision making with reinforcement learning. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2002*, 259-268.

Sun, P. (2003). *Constructing learning models from data: The dynamic catalog mailing problem.* Ph. D. Dissertation, Tsinghua University, China.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, USA.

Zhao, S. Z., Reynolds, M. W., Leikowith, J., Whelton, A., & Arellano, F. M. (2001). A comparison of renal-related adverse drug reactions between rofecoxib and celecoxib, based on World Health Organization/ Uppsala Monitoring Centre safety database. *Clinical Therapeutics*, 23 (9), 1478-1491.

Zukerman, I., & Albrecht, D. (2001). Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11, 5-18.

A

KEY TERMS

Agent: In RL terms, it is the responsible of making decisions according to observations of its environment.

Environment: In RL terms, it is every external condition to the agent.

Exploration-Explotation Dilemma: It is a classical RL dilemma, in which a trade-off solution must be achieved. Exploration means random search of new actions in order to achieve a likely (but yet unknown) better reward than all the known ones, while explotation is focused on exploiting the current knowledge for the maximization of the reward (*greedy* approach).

Life-Time Value: It is a measure widely used in marketing applications that offers the long-term result that has to be maximized.

Reward: In RL terms, the immediate reward is the value returned by the environment to the agent depending on the taken action. The long-term reward is the sum of all the immediate rewards throughout a complete decision process.

Sensitivity: Similar measure that offers the ratio of positives that are correctly classified by the model. (Refer to Specificity.)

Specificity: Success rate measure in a classification problem. If there are two classes (namely, positive and negative), specificity measures the ratio of negatives that are correctly classified by the model.

Algorithms for Association Rule Mining

Vasudha Bhatnagar
University of Delhi, India

Anamika Gupta
University of Delhi, India

Naveen Kumar
University of Delhi, India

INTRODUCTION

Association Rule Mining (ARM) is one of the important data mining tasks that has been extensively researched by data-mining community and has found wide applications in industry. An Association Rule is a pattern that implies co-occurrence of events or items in a database. Knowledge of such relationships in a database can be employed in strategic decision making in both commercial and scientific domains.

A typical application of ARM is market basket analysis where associations between the different items are discovered to analyze the customer's buying habits. The discovery of such associations can help to develop better marketing strategies. ARM has been extensively used in other applications like spatial-temporal, health care, bioinformatics, web data etc (Hipp J., Güntzer U., Nakhaeizadeh G. 2000).

An association rule is an implication of the form $X \rightarrow Y$ where X and Y are independent sets of attributes/items. An association rule indicates that if a set of items X occurs in a transaction record then the set of items Y also occurs in the same record. X is called the antecedent of the rule and Y is called the consequent of the rule. Processing massive datasets for discovering co-occurring items and generating interesting rules in reasonable time is the objective of all ARM algorithms. The task of discovering co-occurring sets of items cannot be easily accomplished using SQL, as a little reflection will reveal. Use of 'Count' aggregate query requires the condition to be specified in the where clause, which finds the frequency of only one set of items at a time. In order to find out all sets of co-occurring items in a database with n items, the number of queries that need to be written is exponential in n. This is the prime motivation for designing algorithms

for efficient discovery of co-occurring sets of items, which are required to find the association rules.

In this article we focus on the algorithms for association rule mining (ARM) and the scalability issues in ARM. We assume familiarity of the reader with the motivation and applications of association rule mining

BACKGROUND

Let $I = \{i_1, i_2, ..., i_n\}$ denote a set of items and D denote a database of N transactions. A typical transaction $T \in D$ may contain a subset X of the entire set of items I and is associated with a unique identifier TID. An *item-set* is a set of one or more items i.e. X is an item-set if $X \subseteq I$. A *k-item-set* is an item-set of cardinality k. A transaction is said to contain an item-set X if $X \subseteq T$. *Support* of an item set X, also called *Coverage* is the fraction of transactions that contain X. It denotes the probability that a transaction contains X.

$$Support(X) = P(X) = \frac{No. \ of \ transactions \ containing \ X}{N}$$

An item-set having support greater than the user specified support threshold (ms) is known as *frequent item-set*.

An *association rule* is an implication of the form $X \rightarrow Y$ [*Support, Confidence*] where $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$, where *Support* and *Confidence* are rule evaluation metrics. *Support* of a rule $X \rightarrow Y$ in D is 'S'' if S% of transactions in D contain $X \cup Y$. It is computed as:

$$Support(X \rightarrow Y) = P(X \cup Y) = \frac{No. of \ transaction \ containing \ X \cup Y}{N}$$

A

Support indicates the prevalence of a rule. In a typical market basket analysis application, rules with very low support values represent rare events and are likely to be uninteresting or unprofitable. *Confidence* of a rule measures its strength and provides an indication of the reliability of prediction made by the rule. A rule $X \rightarrow Y$ has a confidence 'C'' in *D* if C % of transactions in *D* that contain *X*, also contain *Y*. Confidence is computed, as the conditional probability of *Y* occuring in a transaction, given *X* is present in the same transaction, i.e.

$$Confidence(X \rightarrow Y) = P(Y/X) = \frac{P(X \cup Y)}{P(X)} = \frac{Support(X \cup Y)}{Support(X)}$$

A rule generated from frequent item-sets is *strong* if its confidence is greater than the user specified confidence threshold (*mc*). Fig. 1 shows an example database of five transactions and shows the computation of support and confidence of a rule.

The objective of Association Rule Mining algorithms is to discover the set of strong rules from a given database as per the user specified *ms* and *mc* thresholds. Algorithms for ARM essentially perform two distinct tasks: (1) Discover frequent item-sets. (2) Generate strong rules from frequent item-sets.

The first task requires counting of item-sets in the database and filtering against the user specified threshold (*ms*). The second task of generating rules from frequent item-sets is a straightforward process of generating subsets and checking for the strength. We describe below the general approaches for finding frequent item-sets in association rule mining algorithms. The second task is trivial as explained in the last section of the article.

APPROACHES FOR GENERATING FREQUENT ITEM-SETS

If we apply a brute force approach to discover frequent item-sets, the algorithm needs to maintain counters for all $2^n - 1$ item-sets. For large values of *n* that are common in the datasets being targeted for mining, maintaining such large number of counters is a daunting task. Even if we assume availability of such large memory, indexing of these counters also presents a challenge. Data mining researchers have developed numerous algorithms for efficient discovery of frequent item-sets.

The earlier algorithms for ARM discovered all frequent item-sets. Later it was shown by three independent groups of researchers (Pasquier N., Bastide Y., Taouil R. & Lakhal L. 1999), (Zaki M.J. 2000), (Stumme G., 1999), that it is sufficient to discover frequent closed item-sets (FCI) instead of all frequent item-sets (FI). FCI are the item-sets whose support is not equal to the support of any of its proper superset. FCI is a reduced, complete and loss less representation of frequent item-sets. Since FCI are much less in number than FI, computational expense for ARM is drastically reduced.

Figure 2 summarizes different approaches used for ARM. We briefly describe these approaches.

Discovery of Frequent Item-Sets

Level-Wise Approach

Level wise algorithms start with finding the item-sets of cardinality one and gradually work up to the frequent item-sets of higher cardinality. These algorithms use anti-monotonic property of frequent item-sets accord-

Figure 1. Computation of support and confidence of a rule in an example database

TID	Items
1	BCD
2	BCDE
3	AC
4	BDE
5	AB

Let ms=40%, mc=70%
Consider the association rule $B \rightarrow D$,
support $(B \rightarrow D)$ = 3/5 = 60%
confidence$(B \rightarrow D)$ = support(B? D)/support(B)
= 3/4 = 75%
The rule $B \rightarrow D$ is a strong rule.

Figure 2. Approaches for ARM algorithms
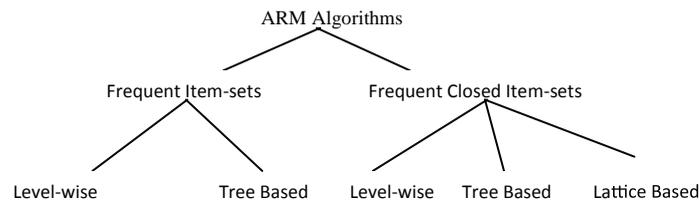
ing to which, no superset of an infrequent item-set can be frequent.

Agarwal et al. (Agarwal, R., Imielinski T., & Swami A. 1993), (Agarwal, R., & Swami A., 1994) proposed Apriori algorithm, which is the most popular iterative algorithm in this category. It starts, with finding the frequent item-sets of size one and goes up level by level, finding candidate item-sets of size *k* by joining item-sets of size *k-1*. Two item-sets, each of size *k-1* join to form an item-set of size *k* if and only if they have first *k-2* items common. At each level the algorithm prunes the candidate item-sets using anti-monotonic property and subsequently scans the database to find the support of pruned candidate item-sets. The process continues till the set of frequent item-sets is non-empty. Since each iteration requires a database scan, maximum number of database scans required is same as the size of maximal item-set. Fig. 3 and Fig 4 gives the pseudo code of Apriori algorithm and a running example respectively.

Two of the major bottlenecks in Apriori algorithm are i) number of passes and ii) number of candidates generated. The first is likely to cause I/O bottleneck and the second causes heavy load on memory and CPU usage. Researchers have proposed solutions to these problems with considerable success. Although detailed discussion of these solutions is beyond the scope of this article, a brief mention is necessary.

Hash techniques reduce the number of candidates by making a hash table and discarding a bucket if it has support less than the *ms*. Thus at each level memory requirement is reduced because of smaller candidate set. The reduction is most significant at lower levels. Maintaining a list of transaction ids for each candidate set reduces the database access. Dynamic Item-set Counting algorithm reduces the number of scans by counting candidate sets of different cardinality in a single scan (Brin S., Motwani R., Ullman J.D., & Tsur S. 1997). Pincer Search algorithm uses a bi-directional strategy to prune the candidate set from top (maximal) and bottom (1-itemset) (Lin D. & Kedem Z.M. 1998). Partitioning and Sampling strategies have also been proposed to speed up the counting task. An excellent comparison of Apriori algorithm and its variants has been given in (Hipp J., Güntzer U., Nakhaeizadeh G. 2000).

Tree Based Algorithms

Tree based algorithms have been proposed to overcome the problem of multiple database scans. These algorithms compress (sometimes lossy) the database into a tree data structure and reduce the number of database scans appreciably. Subsequently the tree is used to mine for support of all frequent item-sets.

Set-Enumeration tree used in Max Miner algorithm (Bayardo R.J. 1998) orders the candidate sets while searching for maximal frequent item-sets. The data structure facilitates quick identification of long frequent item-sets based on the information gathered during each pass. The algorithm is particularly suitable for dense databases with maximal item-sets of high cardinality.

Han et. al. (Han, J., Pei, J., & Yin, Y. 2000) proposed Frequent Pattern (FP)-growth algorithm which performs a database scan and finds frequent item-sets of cardinality one. It arranges all frequent item-sets in a table (header) in the descending order of their supports. During the second database scan, the algorithm constructs in-memory data structure called FP-Tree by inserting each transaction after rearranging it in descending order of the support. A node in FP-Tree stores a single attribute so that each path in the tree

A

Figure 3. Apriori algorithm

```
Input: Database D of N transactions
       ms, mc
Output: Set L of frequent item-sets

Procedure

1.  scan the database and find L₁ = {frequent 1-item sets}
2.  k = 2
3.  while L_{k-1} ≠ φ
4.      C_k = gen_candidate (L_{k-1})
5.      L_k = prune(C_k,L_{k-1})
6.      k++
7.  return L = ∪_k L_k

gen_candidate(L_{k-1})
1.  for each l₁ ∈ L_{k-1}
2.  for each l₂ ∈ L_{k-1}
3.  if(l₁[1] = l₂[1]) ∧ (l₁[2] = l₂[2]) ∧ ...∧ (l₁[k-2] = l₂[k-2]) ∧
        (l₁[k-1] < l₂[k-1]) then
4.          C_k = l₁[1] l₁[2] ... l₁[k-2] l₁[k-1] l₂[k-1]

Prune (C_k , L_{k-1})
// remove candidate itemsets having infrequent subsets
1.  for c ∈ C_k
2.  for each (k-1) subset s of c
3.  if s ∉ L_{k-1} then
4.      remove c from C_k
5.  for each c ∈ C_k
6.      scan the database to find support of c
7.      add c to L_k if support(c) ≥ ms
8.  return L_k
```

represents and counts the corresponding record in the database. A link from the header connects all the nodes of an item. This structural information is used while mining the FP-Tree. FP-Growth algorithm recursively generates sub-trees from FP-Trees corresponding to each frequent item-set.

Coenen et. al. (Coenen F., Leng P., & Ahmed S. 2004) proposed Total Support Tree (T-Tree) and Partial Support Tree (P-Tree) data structures which offer significant advantage in terms of storage and execution. These data structures are compressed set enumeration trees and are constructed after one scan of the database and stores all the item-sets as distinct records in database.

Discovery of Frequent Closed Item-Sets

Level Wise Approach

Pasquier et. al. (Pasquier N., Bastide Y., Taouil R. & Lakhal L. 1999) proposed Close method to find Frequent Closed Item-sets (FCI). This method finds closures based on Galois closure operators and computes the generators. Galois closure operator $h(X)$ for some $X \subseteq I$ is defined as the intersection of transactions in D containing item-set X. An item-set X is a closed item-set if and only if $h(X) = X$. One of the smallest arbitrarily chosen item-set p, such that $h(p) = X$ is known as generator of X.

Close method is based on Apriori algorithm. It starts from 1- item-sets, finds the closure based on Galois closure operator, goes up level by level computing generators and their closures (i.e. FCI) at each level. At each level, candidate generator item-sets of size k are found by joining generator item-sets of size $k-1$ using the combinatorial procedure used in Apriori algorithm. The candidate generators are pruned using two strategies i) remove candidate generators whose all subsets are not frequent ii) remove the candidate generators if closure of one of its subsets is superset of the generator. Subsequently algorithm finds the support of pruned candidate generator. Each iteration requires

Figure 4. Running example of apriori algorithm for finding frequent itemsets (ms = 40%)

Iteration 1

Find 1-itemset

Itemset	Sup(%)
{A}	40
{B}	80
{C}	60
{D}	40
{E}	40

Pruned Item-sets

Itemset	Sup(%)
{A}	40
{B}	80
{C}	60
{D}	40
{E}	40

Iteration 2

Find 2-itemsets

Itemset	Sup(%)
{AB}	20
{AC}	20
{AD}	0
{AE}	0
{BC}	40
{BD}	60
{BE}	40
{CD}	40
{CE}	20
{DE}	40

Pruned itemsets

Itemset	Sup(%)
{BC}	40
{BD}	60
{BE}	40
{CD}	40
{DE}	40

Iteration 3

Find 3-itemsets

Itemset	Sup(%)
{BCD}	40
{BCE}	20
{BDE}	40

Pruned itemsets

Itemset	Sup(%)
{BCD}	40
{BDE}	40

Iteration 4: No candidate item-sets

one pass over the database to construct the set of FCI and count their support.

Tree Based Approach

Wang et. al. (Wang J., Han J. & Pei J. 2003) proposed Closet+ algorithm to compute FCI and their supports using FP-tree structure. The algorithm is based on divide and conquers strategy and computes the local frequent items of a certain prefix by building and scanning its projected database.

Concept Lattice Based Approach

Concept lattice is a core structure of *Formal Concept Analysis* (*FCA*). *FCA* is a branch of mathematics based on *Concept* and *Concept hierarchies*. *Concept* (*A,B*) is defined as a pair of set of objects *A* (known as *extent*) and set of attributes *B* (known as *intent*) such that set of all attributes belonging to *extent A* is same as *B* and set of all objects containing attributes of *intent B* is same as *A*. In other words, no object other than objects of set *A* contains all attributes of *B* and no attribute other than attributes in set *B* is contained in all objects of set *A*. *Concept lattice* is a complete lattice of all *Concepts*. Stumme G., (1999) discovered that *intent*

Exhibit A.

```
add extent {all transactions} in the list of extents
For each item i ∈ I
        for each set X in the list of extents
                find X ∩ {set of transactions containing i}
                include in the list of extents if not included earlier
        EndFor
EndFor
```

B of the *Concept* (*A,B*) represents the closed item-set, which implies that all algorithms for finding Concepts can be used to find closed item-sets. Kuznetsov S.O., & Obiedkov S.A. (2002) provides a comparison of performance of various algorithms for concepts. The naïve method to compute Concepts, proposed by Ganter is given in Exhibit A.

This method generates all the *Concepts* i.e. all closed item-sets. Closed item-sets generated using this method in example 1 are *{A},{B} ,{C},{A,B},{A,C},{B,D},{B, C,D}, {B,D,E}, {B,C,D,E}*. Frequent Closed item-sets are *{A} ,{B},{C},{B,D},{B,C,D},{B,D,E}*.

Concept lattice for frequent closed item-sets is given in Figure 5.

Generating Association Rules

Once all frequent item-sets are known, association rules can be generated in a straightforward manner by finding all subsets of an item-sets and testing the strength (Han J., & Kamber M., 2006). The pseudo code for this algorithm is given in Exhibit B.

Based on the above algorithm, *strong* rules generated from frequent item-set BCD in Example 1 are:

$BC \rightarrow D,$ conf=100%
$CD \rightarrow B,$ conf=100%
where $mc = 70\%$

There are two ways to find association rules from frequent closed item-sets:

i)   compute frequent item-sets from FCI and then find the association rules
ii) generate rules directly using FCI.

Close method uses the first approach, which generates lot of redundant rules while method proposed by Zaki (Zaki M.J., 2000), (Zaki, M.J., & Hsiao C., J., 2005) uses the second approach and derives rules

Figure 5. Concept lattice

directly from the *Concept lattice*. The association rules thus derived are *non-redundant rules*. For example, set of *strong* rules generated using Close method in Example 1 is {BC → D,CD →B,D →B,E → B,E →D,E → BD, BE →D,DE →B}. For the same example, set of non-redundant strong rules generated using *Concept Lattice* approach is {D →B, E → BD, BC → D, CD → B}. We can observe here that all rules can be derived from the reduced non-redundant set of rules.

Scalability issues in Association Rule Mining

Scalability issues in ARM have motivated development of incremental and parallel algorithms. Incremental algorithms for ARM preserve the counts of selective item-sets and reuse this knowledge later to discover frequent item-sets from augmented database. Fast update algorithm (FUP) is the earliest algorithm based on this idea. Later different algorithms are presented based on sampling (Hipp J., Guntzer U., & Nakhaeizadeh G., 2000).

Parallel algorithms partition either the dataset for counting or the set of counters, across different ma-

Exhibit B.

For each frequent item-set *I*,
    generate all non-empty subsets of *I*
    For every non-empty subset *s* of *I*,
        Output the rule $s \to (I\text{-}s)$ if support(*I*) / support (*s*) >= *mc*
    EndFor
EndFor

chines to achieve scalability (Hipp J., Guntzer U., & Nakhaeizadeh G., 2000). Algorithms, which partition the dataset exchange counters while the algorithms, which partition the counters, exchange datasets incurring high communication cost.

FUTURE TRENDS

Discovery of Frequent Closed Item-sets (FCI) is a big lead in ARM algorithms. With the current growth rate of databases and increasing applications of ARM in various scientific and commercial applications we envisage tremendous scope for research in parallel, incremental and distributed algorithms for FCI. Use of lattice structure for FCI offers promise of scalability. On line mining on streaming datasets using FCI approach is an interesting direction to work on.

CONCLUSION

The article presents the basic approach for Association Rule Mining, focusing on some common algorithms for finding frequent item-sets and frequent closed item-sets. Various approaches have been discussed to find such item-sets. Formal Concept Analysis approach for finding frequent closed item-sets is also discussed. Generation of rules from frequent items-sets and frequent closed item-sets is briefly discussed. The article addresses the scalability issues involved in various algorithms.

REFERENCES

Agarwal, R., Imielinski T., & Swami A., (1993), Mining Association Rules Between Sets of Items in Large Databases, Proceedings of the 1993 ACM International Conference on Management of Data, 207-216, Washington, D.C.

Agrawal R., & Srikant R., (1994), Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the Twentieth International Conference on VLDB, pp. 487-499, Santiago, Chile

Bayardo R.J. (1998), **Efficiently Mining Long Patterns From Databases,** Proceedings of the ACM International Conference on Management of Data.

Brin S., Motwani R., Ullman J. D., & Tsur S., (1997), Dynamic Item-set Counting and Implication Rules for Market Basket Data. ACM Special Interest Group on Management of Data, 26(2):255

Coenen F., Leng P., & Ahmed S., (2004) Data Structure for Association Rule Mining: T-Trees and P-Trees, IEEE TKDE, Vol. 16, No. 6

Han, J., Pei, J., & Yin, Y., (2000), Mining Frequent Patterns Without Candidate Generation, Proceedings of the ACM International Conference on Management of Data, ACM Press, 1-12.

Han, J., & Kamber, M., (2006), Data Mining: Concepts and Techniques, 2nd ed. Morgan Kaufmann Publishers.

Hipp, J., Güntzer, U., & Nakhaeizadeh, G., (2000), Algorithms for Association Rule Mining: A General Survey and Comparison, SIGKDD Explorations.

Kuznetsov, S.O., & Obiedkov, S.A., (2002), Comparing Performance of Algorithms For Generating Concept Lattices, Journal of Experimentation and Theoretical Artificial Intelligence.

Lin, D., & Kedem, Z. M., (1998), Pincer Search: A New Algorithm for Discovering the Maximum Frequent

A

Sets. Proceedings of the 6th International Conference on Extending Database Technology (EDBT), Valencia, Spain.

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L., (1999), Efficient Mining of Association Rules Using Closed Item-set Lattices, Information Systems, 24(1):25-46

Stumme, G., (1999), Conceptual Knowledge Discovery with Frequent Concept Lattices, FB4-Preprint 2043, TU Darmstadt

Wang, J., Han, J., & Pei, J., (2003), Closet+: Searching for the Best Strategies for Mining Frequent Closed Item-sets, Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 236-245, New York, USA, ACM Press.

Zaki, M. J., (2000), Generating Non-Redundant Association Rules, Proceedings of the International Conference on Knowledge Discovery and Data Mining.

Zaki, M.J., & Hsiao C.,J.,(2005), Efficient algorithms for mining closed item-sets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering, 17(4): 462-478.

KEY TERMS

Association Rule: An Association rule is an implication of the form $X \rightarrow Y$ where $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$, I denotes the set of items.

Data Mining: Extraction of interesting, non-trivial, implicit, previously unknown and potentially useful information or patterns from data in large databases.

Formal Concept: A formal context $K = (G,M,I)$ consists of two sets G (objects) and M (attributes) and a relation I between G and M. For a set $A \subseteq G$ of objects

$A' = \{m \varepsilon M \mid gIm \text{ for all } g \varepsilon A\}$ (the set of all attributes common to the objects in A). Correspondingly, for a set B of attributes we define

$B' = \{g \varepsilon G \mid gIm \text{ for all } m \varepsilon B\}$ (the set of objects common to the attributes in B).

A formal concept of the context (G,M,I) is a pair (A,B) with $A \subseteq G, B \subseteq M,$

$A' = B$ and $B' = A$

A is called the extent and B is the intent of the concept (A,B).

Frequent Closed Item-Set: An item-set X is a closed item-set if there exists no item-set X' such that:

i. X' is a proper superset of X,
ii. Every transaction containing X also contains X'.

A closed item-set X is frequent if its support exceeds the given support threshold.

Galois Connection: Let $D = (\mathsf{O}, \mathsf{I}, \mathsf{R})$ be a data mining context where O and I are finite sets of objects (transactions) and items respectively. $\mathsf{R} \subseteq \mathsf{O} \times \mathsf{I}$ is a binary relation between objects and items. For $O \subseteq \mathsf{O}$, and $I \subseteq \mathsf{I}$, we define as shown in Exhibit C.

$f(O)$ associates with O the items common to all objects $o \in O$ and $g(I)$ associates with I the objects related to all items $i \in I$. The couple of applications (f,g) is a Galois connection between the power set of O (i.e. 2^{O}) and the power set of I (i.e. 2^{I}).

The operators $h = f \circ g$ in 2^{I} and $h' = g \circ f$ in 2^{O} are Galois closure operators. An item-set $C \subseteq \mathsf{I}$ from D is a closed item-set iff $h(C) = C$.

Generator Item-Set: A generator p of a closed item-set c is one of the smallest item-sets such that $h(p) = c$.

Non-Redundant Association Rules: Let R_i denote the rule $X_1^i \rightarrow X_2^i$, where $X_1, X_2 \subseteq I$. Rule R_1 is more general than rule R_2 provided R_2 can be generated by adding additional items to either the antecedent or consequent of R_1. Rules having the same support and confidence as

Exhibit C.

$$f(O): 2^{\mathsf{O}} \rightarrow 2^{\mathsf{I}}$$
$$f(O) = (i \in I \mid \forall o \in \mathsf{O}, (o,i) \in \mathsf{R})$$

$$g(I): 2^{\mathsf{I}} \rightarrow 2^{\mathsf{O}}$$
$$g(I) = (o \in O \mid \forall i \in I, (o,i) \in \mathsf{R})$$

more general rules are the redundant association rules.
Remaining rules are non-redundant rules.

Ambient Intelligence

Fariba Sadri
Imperial College London, UK

Kostas Stathis
Royal Holloway, University of London, UK

INTRODUCTION

In recent years much research and development effort has been directed towards the broad field of **ambient intelligence (AmI)**, and this trend is set to continue for the foreseeable future. AmI aims at seamlessly integrating services within smart infrastructures to be used at home, at work, in the car, on the move, and generally in most environments inhabited by people. It is a relatively new paradigm rooted in ubiquitous computing, which calls for the integration and convergence of multiple disciplines, such as sensor networks, portable devices, intelligent systems, human-computer and social interactions, as well as many techniques within artificial intelligence, such as planning, contextual reasoning, speech recognition, language translation, learning, adaptability, and temporal and hypothetical reasoning.

The term AmI was coined by the European Commission, when in 2001 one of its Programme Advisory Groups launched the AmI challenge (Ducatel et al., 2001), later updated in 2003 (Ducatel et al., 2003). But although the term AmI originated from Europe, the goals of the work have been adopted worldwide, see for example (The Aware Home, 2007), (The Oxygen Project, 2007), and (The Sony Interaction Lab, 2007).

The foundations of AmI infrastructures are based on the impressive progress we are witnessing in wireless technologies, sensor networks, display capabilities, processing speeds and mobile services. These developments help provide much useful (row) information for AmI applications. Further progress is needed in taking full advantage of such information in order to provide the degree of intelligence, flexibility and naturalness envisaged. This is where **artificial intelligence** and **multi-agent techniques** have important roles to play.

In this paper we will review the progress that has been made in intelligent systems, discuss the role of artificial intelligence and **agent technologies** and focus on the application of AmI for independent living.

BACKGROUND

Ambient intelligence is a vision of the information society where normal working and living environments are surrounded by **embedded intelligent devices** that can merge unobtrusively into the background and work through intuitive interfaces. Such devices, each specialised in one or more capabilities, are intended to work together within an infrastructure of **intelligent systems**, to provide a multitude of services aimed at generally improving safety and security and improving quality of life in ordinary living, travelling and working environments.

The European Commission identified four AmI scenarios (Ducatel et al. 2001, 2003) in order to stimulate imagination and initiate and structure research in this area. We summarise two of these to provide the flavour of AmI visions.

AmI Scenarios:

1. Dimitrios is taking a coffee break and prefers not to be disturbed. He is wearing on his clothes or body a voice activated digital avatar of himself, known as Digital Me (D-Me). D-Me is both a learning device, learning about Dimitrios and his environment, and an acting device offering communication, processing and decision-making functionalities. During the coffee break D-Me answers the incoming calls and emails of Dimitrios. It does so smoothly in the necessary languages, with a re-production of Dimitrios' voice and accent. Then D-Me receives a call from Dimitrios' wife, recognises its urgency and passes it on to Demetrios. At the same time it catches a message from an older person's D-Me,

located nearby. This person has left home without his medication and would like to find out where to access similar drugs. He has asked his D-Me, in natural language, to investigate this. Dimitrios happens to suffer from a similar health problem and uses the same drugs. His D-Me processes the incoming request for information, and decides neither to reveal Dimitrios' identity nor offer direct help, but to provide the elderly person's D-Me with a list of the closest medicine shops and potential contact with a self-help group.

2. Carmen plans her journey to work. It asks AmI, by voice command, to find her someone with whom she can share a lift to work in half an hour. She then plans the dinner party she is to give that evening. She wishes to bake a cake, and her e-fridge flashes a recipe on the e-fridge screen and highlights the ingredients that are missing. Carmen completes her shopping list on the screen and asks for it to be delivered to the nearest distribution point in her neighbourhood. All goods are smart tagged, so she can check the progress of her virtual shopping from any enabled device anywhere, and make alterations. Carmen makes her journey to work, in a car with dynamic traffic guidance facilities and traffic systems that dynamically adjust speed limits depending on congestion and pollution levels. When she returns home the AmI welcomes her and suggests that on the next day she should telework, as a big demonstration is planned in downtown.

The demands that drive AmI and provide opportunities are for improvement of safety and quality of life, enhancements of productivity and quality of products and services, including public services such as hospitals, schools, military and police, and industrial innovation. AmI is intended to facilitate human contact and community and cultural enhancement, and ultimately it should inspire trust and confidence.

Some of the technologies required for AmI are summarised in Figure 1.

AmI work builds on ubiquitous computing and sensor network and mobile technologies. To provide the intelligence and naturalness required, it is our view that significant contributions can come from advances in artificial intelligence and agent technologies. **Artificial intelligence** has a long history of research on planning, scheduling, temporal reasoning, fault diagnosis, hypothetical reasoning, and reasoning with incomplete and uncertain information. All of these are techniques that can contribute to AmI where actions and decisions have to be taken in real time, often with dynamic and uncertain knowledge about the environment and the user. Agent technology research has concentrated on agent architectures that combine several, often cognitive, capabilities, including reactivity and adaptability, as well as the formation of agent societies through communication, norms and protocols.

Recent work has attempted to exploit these techniques for AmI. In (Augusto and Nugent 2004) the use of temporal reasoning combined with active data-

Figure 1. Components of Ambient Intelligence

86
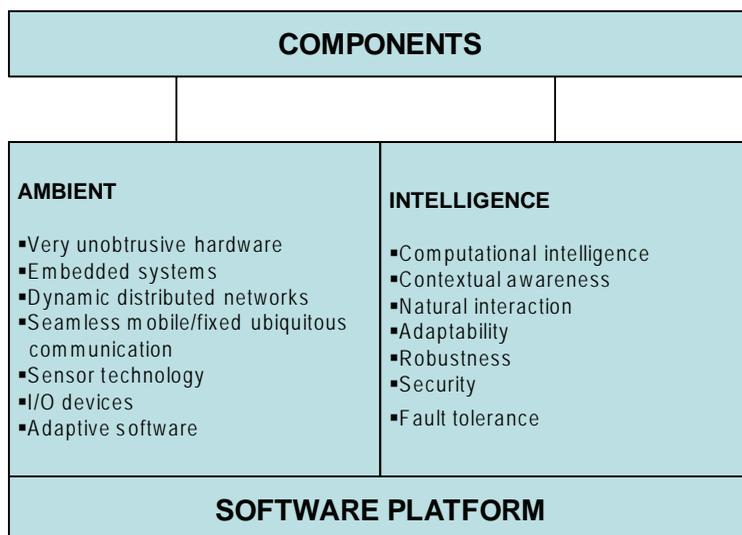
bases are explored in the context of smart homes. In (Sadri 2007) the use of temporal reasoning together with agents is explored to deal with similar scenarios, where information observed in a home environment is evaluated, deviations from normal behaviour and risky situations are recognised and compensating actions are recommended.

The relationship of AmI to **cognitive agents** is motivated by (Stathis and Toni 2004) who argue that computational logic elevates the level of the system to that of a user. They advocate the **KGP agent model** (Kakas, et al 2004) to investigate how to assist a traveller to act independently and safely in an unknown environment using a personal communicator. (Augusto et al 2006) address the process of taking decisions in the presence of conflicting options. (Li and Ji 2005) offer a new probabilistic framework based on Bayesian Networks for dealing with ambiguous and uncertain sensory observations and users' changing states, in order to provide correct assistance.

(Amigoni et al 2005) address the goal-oriented aspect of AmI applications, and in particular the planning problem within AmI. They conclude that a combination of centralised and distributed planning capabilities are required, due to the distributed nature of AmI and the participation of heterogeneous agents, with different capabilities. They offer an approach based on the Hierarchical Task Networks taking the perspective of a multi-agent paradigm for AmI.

The paradigm of **embedded agents** for AmI environments with a focus on developing learning and adaptation techniques for the agents is discussed in (Hagras et al 2004, and Hagras and Callaghan 2005). Each agent is equipped with sensors and effectors and uses a learning system based on fuzzy logic. A real AmI environment in the form of an "intelligent dormitory" is used for experimentation.

Privacy and security in the context of AmI applications at home, at work, and in the health, shopping and mobility domains are discussed in (Friedewald et al 2007). For such applications they consider security threats such as surveillance of users, identity theft and malicious attacks, as well as the potential of the digital divide amongst communities and social pressures.

AMBIENT INTELLIGENCE FOR INDEPENDENT LIVING

A

One major use of AmI is to support services for **independent living**, to prolong the time people can live decently in their own homes by increasing their autonomy and self-confidence. This may involve the elimination of monotonous everyday activities, monitoring and caring for the elderly, provision of security, or saving resources. The aim of such AmI applications is to help:

- maintain safety of a person by monitoring his environment and recognizing and anticipating risks, and taking appropriate actions,
- provide assistance in daily activities and requirements, for example, by reminding and advising about medication and nutrition, and
- improve quality of life, for example by providing personalized information about entertainment and social activities.

This area has attracted a great deal of attention in recent years, because of increased longevity and the aging population in many parts of the world. For such an AmI system to be useful and accepted it needs to be versatile, adaptable, capable of dealing with changing environments and situations, transparent and easy, and even pleasant, to interact with.

We believe that it would be promising to explore an approach based on providing an **agent architecture** consisting of a society of heterogeneous, intelligent, **embedded agents**, each specialised in one or more functionalities. The agents should be capable of sharing information through communication, and their dialogues and behaviour should be governed by context-dependent and dynamic norms.

The basic capabilities for intelligent agents include:

- *Sensing*: to allow the agent observe the environment
- *Reactivity*: to provide context-dependent dynamic behaviour and the ability to adapt to changes in the environment
- *Planning*: to provide goal-directed behaviour
- *Goal Decision*: to allow dynamic decisions about which goals have higher priorities

- *Action execution*: to allow the agent to affect the environment.

All of these functionalities also require reasoning about spatio-temporal constraints reflecting the environment in which an AmI system operates.

Most of these functionalities have been integrated in the **KGP model** (Kakas et al, 2004), whose architecture is shown in Figure 2 and implemented in the PROSOCS system (Bracciali et al, 2006). The use of **reactivity** for communication and dialogue policies has also been discussed in, for example, (Sadri et al, 2003). The inclusion of normative behaviour has been discussed in (Sadri et al, 2006) where we also consider how to choose amongst different types of goals, depending on the governing norms. For a general discussion on the importance of norms in artificial societies see (Pitt, 2005).

KGP agents are situated in the environment via their *physical capabilities*. Information received from the environment (including other agents) updates the agents *state* and provides input to its dynamic *cycle theory,* which, in turn, determines the next steps in terms of its *transitions*, using its ***reasoning capabilities***.

FUTURE TRENDS

As most other information and communication technologies, AmI is not likely to be good or bad on its own, but its value will be judged from the different

Figure 2. The architecture of a KGP agent

ways the technology will be used to improve people's lives. In this section we discuss new opportunities and challenges for the integration of AmI with what people do in ordinary settings. We abstract away from hardware trends and we focus on areas that are software related and are likely to play an important role in the adoption of AmI technologies.

A focal point is the observation that people discover and understand the world through visual and conversational interactions. As a result, in the coming years we expect to see the design of AmI systems to focus in ways that will allow humans to interact in natural ways, using their common skills such as speaking, gesturing, glancing. This kind of *natural interaction* (Leibe et al 2000) will complement existing interfaces and will require that AmI systems be capable of representing virtual objects, possibly in 3D, as well as capture people's moves in the environment and identify which of these moves are directed to virtual objects.

We also expect to see new research directed towards processing of sensor data with different information (Massaro and Friedman 1990) and different kind of formats such as audio, video, and RFID. Efficient techniques to index, search, and structure these data and ways to transform them to the higher-level semantic information required by cognitive agents will be an important area for future work. Similarly, the reverse of this process is likely to be of equal importance, namely, how to translate high-level information to the lower-level signals required by actuators that are situated in the environment.

Given that sensors and actuators will provide the link with the physical environment, we also anticipate further research to address the general linking of AmI systems to already existing computing infrastructures such as the *semantic web*. This work will create hybrid environments that will need to combine useful information from existing wired technologies with information from wireless ones (Stathis et al 2007). To enable the creation of such environments we imagine the need to build new frameworks and middleware to facilitate integration of heterogeneous AmI systems and make the interoperation more flexible.

Another important issue is how the human experience in AmI will be managed in a way that will be as unobtrusive as possible. In this we foresee that developments in cognitive systems will play a very important role. Although there will be many areas of cognitive system behaviour that will need to be addressed, we
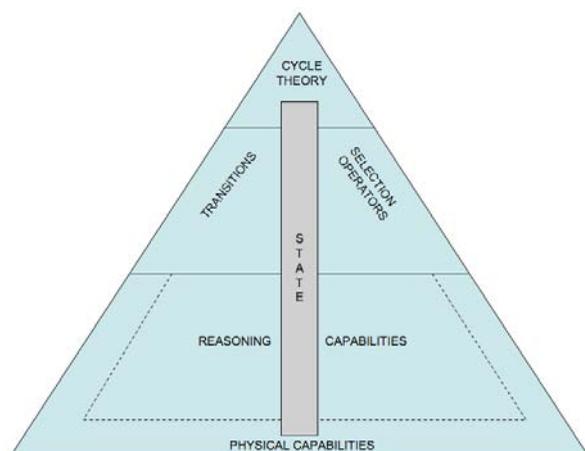
anticipate that development of **agent models** that adapt and learn (Sutton and Barto 1998), to be of great importance. The challenge here will be how to integrate the output of these adaptive and learning capabilities to the reasoning and decision processes of the agent. The resulting cognitive behaviour must differentiate between newly learned concepts and existing ones, as well as discriminate between normal behaviour and exceptions.

We expect that AmI will emerge with the formation of user communities who live and work in a particular locality (Stathis et al 2006). The issue then becomes how to manage all the information that is provided and captured as the system evolves. We foresee research to address issues such as semantic annotations of content, and partitioning and ownership of information.

Linking in local communities with smart homes, e-healthcare, mobile commerce, and transportation systems will eventually give rise to a global AmI system. For applications in such a system to be embraced by people we will need to see specific human factors studies to decide how unobtrusive, acceptable and desirable the actions of the AmI environment seem to people who use them. Some human factors studies should focus on issues of presentation of objects and agents in a 3D setting, as well as on the important issues of privacy, trust and security.

To make possible the customization of system interactions to different classes of users, it is required to acquire and store information about these users. Thus for people to trust AmI interactions in the future we must ensure that the omnipresent intelligent environment maintains privacy in an ethical manner. Ethical or, better, normative behaviour cannot only be ensured at the cognitive level (Sadri et al 2006), but also at the lower, implementation level of the AmI platform. In this context, ensuring that communicated information is encrypted, certified, and follows transparent security policies will be required to build systems less vulnerable to malicious attacks. Finally, we also envisage changes to business models that would characterise AmI interactions (Hax and Wielde 2001).

CONCLUSION

The successful adoption of AmI is predicated on the suitable combination of ubiquitous computing, artificial intelligence and agent technologies. A useful class of applications that can test such a combination is AmI supporting independent living. For such applications we have identified the trends that are likely to play an important role in the future.

REFERENCES

Augusto, J.C., Liu, J., Chen L. (2006). Using ambient intelligence for disaster management. In the Proceedings of the 10th International Conference on Knowledge-based Intelligent Information and Engineering Systems (KES 2006), Springer Verlag.

Augusto, J.C., Nugent, C. D. (2004). The use of temporal reasoning and management of complex events in smart homes. In Proceedings of the European Conference on Artificial Systems (ECAI), 778-782.

Bracciali, A., Endriss, U., Demetriou, N., Kakas, A.C., Lu, L., Stathis, K. (2006). Crafting the mind of PROSOCS agents. Applied Artificial Intelligence 20(2-4), 105-131.

Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman J.-C. (2001). Scenarios for ambient intelligence in 2010. IST Advisory Group Final Report, European Commission.

Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman J.-C. (2003). Ambient intelligence : from vision to reality. IST Advisory Group Draft Report, European Commission.

Dutton, W. H. (1999). Society on the line: information politics in the digital age, Oxford, Oxford University Press.

Friedewald M., Vildijiounaite, E., Punie, Y. Wright, D. (2007). Privacy, identity and security in ambient intelligence: a scenario analysis. Telematics and Informatics, 24, 15-29.

Hagras, H., Callaghan, V., Colley, M., Clarke, G., Pounds-Cornish, A., Duman, H. (2004). Creating an ambient intelligence environment using embedded agents. IEEE Intelligent Systems, 19(6), 12-20.

Hagras, H. and Callaghan, V. (2005). An intelligent fuzzy agent approach for realizing ambient intelligence in intelligent inhabited environments. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 35(1), 55-65.

A

Hax, A., and Wilde, D, II. (2001). The Delta Model – discovering new sources of profitability in a networked economy. European Management Journal. 9, 379-391.

Kakas, A., Mancarella, P., Sadri, F. Stathis, K. Toni, F. (2004). The KGP model of agency. In Proceedings of European Conference on Artificial Intelligence, 33-37.

Li, X. and Ji, Q. (2005). Active affective state detection and user assistance with dynamic bayesian networks. IEEE Transactions on Systems, Man, and Cybernetics: Special Issue on Ambient Intelligence, 35(1), 93-105.

Leibe, B., Starner, T., Ribarsky, W., Wartell, Z., Krum, D., Singletary, B., and Hodges, L. (2000). The Perceptive Workbench: towards spontaneous and natural interaction in semi-immersive virtual environments. IEEE Virtual Reality Conference (VR'2000). 13-20.

Massaro, D. W., and D. Friedman. (1990). Models of integration given multiple sources of information. Psychological Review. 97, 225-252.

Pitt, J. (2005) The open agent society as a platform for the user-friendly information society. AI Soc. 19(2), 123-158.

Sadri, F., Stathis, K., and Toni, F. (2006). Normative KGP agents. Journal of Computational and Mathematical Organizational Theory. 12(2-3), 101-126.

Sadri, F. (2007). Ambient intelligence for care of the elderly in their homes. In Proceedings of the 2nd Workshop on Artificial Intelligent Techniques for Ambient Intelligence (AITAmI '07), 62-67.

Sadri, F., Toni, F., Torroni, P. (2003). Minimally intrusive negotiating agents for resource sharing. In Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI 03), 796-801.

Stathis, K., de Bruijn, O., Spence, R. and Purcell, P. (2006) Ambient intelligence: human-agent interactions in a networked community. In Purcell, P. (ed) Networked Neighbourhoods: The Connected Community in Context (Springer), 279-304.

Stathis, K., Kafetzoglou, S., Papavasiliou, S., and Bromuri, S. (2007). Sensor network grids: agent environments combined with QoS in wireless sensor networks. In Proceedings of the 3rd International Conference on Autonomic and Autonomous Systems, IEEE. 47-52.

Stathis, K. And Toni, F. (2004). Ambient intelligence using KGP agents. Workshop at the Second European Symposium on Ambient Intelligence, Lecture Notes in Compuer Science 3295, 351-362.

Sutton, R. S. and Barto, G. A. (1998). Reinforcement learning: an introduction. MIT Press.

The Aware Home Initiative (2007), http://www.cc.gatech.edu/fce/house/house.html.

The Oxygen Project (2007), http://www.oxygen.lcs.mit.edu.

The Sony Interaction Lab (2007), http://www.sonycsl.co.jp/IL/index.html.

TERMS AND DEFINITIONS

Artificial Societies: Complex systems consisting of a, possibly large, set of agents whose interaction are constrained by norms and the roles the agents are responsible to play.

Cognitive Agents: Software agents endowed with high-level mental attitudes, such as beliefs, goals and plans.

Context Awareness: Refers to the idea that computers can both sense and react according to the state of the environment they are situated. Devices may have information about the circumstances under which they are able to operate and react accordingly.

Natural Interaction: The investigation of the relationships between humans and machines aiming to create interactive artifacts that respect and exploit the natural dynamics through which people communicate and discover the real world.

Smart Homes: Homes equipped with intelligent sensors and devices within a communications infrastructure that allows the various systems and devices to communicate with each other for monitoring and maintenance purposes.

Ubiquitous Computing: A model of human-computer interaction in which information processing is integrated into everyday objects and activities. Unlike the desktop paradigm, in which a single user chooses to interact with a single device for a specialized purpose, with ubiquitous computing a user interacts with many computational devices and systems simultaneously, in the course of ordinary activities, and may not necessarily even be aware that is doing so.

Wireless Sensor Networks: Wireless networks consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations.

Ambient Intelligence Environments

Carlos Ramos
Polytechnic of Porto, Portugal

INTRODUCTION

The trend in the direction of hardware cost reduction and miniaturization allows including computing devices in several objects and environments (embedded systems). Ambient Intelligence (AmI) deals with a new world where computing devices are spread everywhere (ubiquity), allowing the human being to interact in physical world environments in an intelligent and unobtrusive way. These environments should be aware of the needs of people, customizing requirements and forecasting behaviours.

AmI environments may be so diverse, such as homes, offices, meeting rooms, schools, hospitals, control centers, transports, touristic attractions, stores, sport installations, and music devices.

Ambient Intelligence involves many different disciplines, like automation (sensors, control, and actuators), human-machine interaction and computer graphics, communication, ubiquitous computing, embedded systems, and, obviously, Artificial Intelligence. In the aims of Artificial Intelligence, research envisages to include more intelligence in the AmI environments, allowing a better support to the human being and the access to the essential knowledge to make better decisions when interacting with these environments.

BACKGROUND

Ambient Intelligence (AmI) is a concept developed by the European Commission's IST Advisory Group ISTAG (ISTAG, 2001)(ISTAG, 2002). ISTAG believes that it is necessary to take a holistic view of Ambient Intelligence, considering not just the technology, but the whole of the innovation supply-chain from science to end-user, and also the various features of the academic, industrial and administrative environment that facilitate or hinder realisation of the AmI vision (ISTAG, 2003). Due to the great amount of technologies involved in the Ambient Intelligence concept we may find several works that appeared even before the ISTAG vision pointing in the direction of Ambient Intelligence trends.

In what concerns Artificial Intelligence (AI), Ambient Intelligence is a new meaningful step in the evolution of AI (Ramos, 2007). AI has closely walked side-by-side with the evolution of Computer Science and Engineering. The building of the first artificial neural models and hardware, with the Walter Pitts and Warren McCullock work (Pitts & McCullock, 1943) and Marvin Minsky and Dean Edmonds SNARC system correspond to the first step. Computer-based Intelligent Systems, like the MYCIN Expert System (Shortliffe, 1976) or network-based Intelligent Systems, like AUTHORIZER's ASSISTANT (Rothi, 1990) used by American Express for authorizing transactions consulting several Data Bases are the kind of systems of the second step of AI. From the 80's Intelligent Agents and Multi-Agent Systems have established the third step, leading more recently to Ontologies and Semantic Web. From hardware to the computer, from the computer to the local network, from the local network to the Internet, and from the Internet to the Web, Artificial Intelligence was on the state of the art of computing, most of times a little bit ahead of the technology limits.

Now the centre is no more in the hardware, or in the computer, or even in the network. Intelligence must be provided to our daily-used environments. We are aware of the push in the direction of Intelligent Homes, Intelligent Vehicles, Intelligent Transportation Systems, Intelligent Manufacturing Systems, even Intelligent Cities. This is the reason why Ambient Intelligence concept is so important nowadays (Ramos, 2007).

Ambient Intelligence is not possible without Artificial Intelligence. On the other hand, AI researchers must be aware of the need to integrate their techniques with other scientific communities' techniques (e.g. Automation, Computer Graphics, Communications). Ambient Intelligence is a tremendous challenge, needing the better effort of different scientific communities.

There is a miscellaneous of concepts and technologies related with Ambient Intelligence. Ubiquitous Computing, Pervasive Computing, Embedded Systems, and Context Awareness are the most common. However these concepts are different from Ambient Intelligence.

The concept of Ubiquitous Computing (UbiComp) was introduced by Mark Weiser during his tenure as Chief Technologist of the Palo Alto Research Center (PARC) (Weiser, 1991). Ubiquitous Computing means that we have access to computing devices anywhere in an integrated and coherent way. Ubiquitous Computing was mainly driven by Communications and Computing devices scientific communities but now is involving other research areas. Ambient Intelligence differs from Ubiquitous Computing because sometimes the environment where Ambient Intelligence is considered is simply local. Another difference is that Ambient Intelligence makes more emphasis on intelligence than Ubiquitous Computing. However, ubiquity is a real need today and Ambient Intelligence systems are considering this feature.

A concept that sometimes is seen as a synonymous of Ubiquitous Computing is Pervasive Computing. According to Teresa Dillon, Ubiquitous Computing is best considered as the underlying framework, the embedded systems, networks and displays which are invisible and everywhere, allowing us to 'plug-and-play' devices and tools, On the other hand, Pervasive Computing, is related with all the physical parts of our lives; mobile phone, hand-held computer or smart jacket (Dillon, 2006).

Embedded Systems mean that electronic and computing devices are embedded in current objects or goods. Today goods like cars are equipped with microprocessors; the same is true for washing machines, refrigerators, and toys. Embedded Systems community is more driven by electronics and automation scientific communities. Current efforts go in the direction to include electronic and computing devices in the most usual and simple objects we use, like furniture or mirrors. Ambient Intelligence differs from Embedded Systems since computing devices may be clearly visible in AmI scenarios. However, there is a clear trend to involve more embedded systems in Ambient Intelligence.

Context Awareness means that the system is aware of the current situation we are dealing with. An example is the automatic detection of the current situation in a Control Centre. Are we in presence of a normal situation or are we dealing with a critical situation, or even an emergency? In this Control Centre the intelligent alarm processor will exhibit different outputs according to the identified situation (Vale, Moura, Fernandes, Marques, Rosado, Ramos, 1997). Automobile Industry is also investing in Context Aware systems, like near-accident detection. Human-Computer Interaction scientific community is paying lots of attention to Context Awareness. Context Awareness is one of the most desired concepts to include in Ambient Intelligence, the identification of the context is important for deciding to act in an intelligent way.

There are different views of the importance of other concepts and technologies in the Ambient Intelligence field. Usually these differences are derived from the basic scientific community of the authors. ISTAG see the technology research requirements from different points of view (Components, Integration, System, and User/Person). In (ISTAG, 2003) the following ambient components are mentioned: smart materials; MEMS and sensor technologies; embedded systems; ubiquitous communications; I/O device technology; adaptive software. In the same document ISTAG refers the following intelligence components: media management and handling; natural interaction; computational intelligence; context awareness; and emotional computing.

Recently Ambient Intelligence is receiving a significant attention from Artificial Intelligence Community. We may refer the Ambient Intelligence Workshops organized by Juan Augusto and Daniel Shapiro at ECAI'2006 (European Conference on Artificial Intelligence) and IJCAI'2007 (International Joint Conference on Artificial Intelligence) and the Special Issue on Ambient Intelligence, coordinated by Carlos Ramos, Juan Augusto and Daniel Shapiro to appear in the March/April'2008 issue of the IEEE Intelligent Systems magazine.

AMBIENT INTELLIGENT PROTOTYPES AND SYSTEMS

Here we will analyse some examples of Ambient Intelligence prototypes and systems, divided by the area of application.

AmI at Home

Domotics is a consolidated area of activity. After the first experiences using Domotics at homes there was a trend to refer the Intelligent Home concept. However, Domotics is too centred in the automation, giving to the user the capability to control the house devices from everywhere. We are still far from the real Ambient Intelligence in homes, at least at the commercial level. In (Wichert, Hellschimidt, 2006) there is an interesting example in the aims of EMBASSI project, by gesture a woman is commanding the TV to be brighter, however the TV is already at the brightest level, so the lights reduce the level and the windows close, showing an example of context awareness in the environment.

Several organizations are doing experiments to achieve the Intelligent Home concept. Some examples are HomeLab from Philips, MIT House_*n*, Georgia Tech Aware Home, Microsoft Concept Home, and e2 Home from Electrolux and Ericsson.

AmI in Vehicles and Transports

Since the first experiences with NAVLAB 1 (Thorpe, Herbert, Kanade, Shafer, 1988) Carnegie Mellon University has developed several prototypes for Autonomous Vehicle Driving and Assistance. The last one, NAVLAB 11, is an autonomous Jeep. Most of the car industry companies are doing research in the area of Intelligent Vehicles for several tasks like car parking assistance or pre-collision detection.

Another example of AmI application is related with Transports, namely in connection with Intelligent Transportation Systems (ITS). The ITS Joint Program of the US Department of Transportation identified several areas of applications, namely: arterial management; freeway management; transit management; incident management; emergence management; electronic payment; traveller information; information management; crash prevention and safety; roadway operations and management; road weather management; commercial vehicle operations; and intermodal freight. In all these application areas Ambient Intelligence can be used.

AmI in Elderly and Health Care

Several studies point to the aging of population during the next decades. While being a good result of increasing of life expectation, this also implies some

problems. The percentage of population with health problems will increase and it will be very difficult to Hospitals to maintain all patients. Our society is faced with the responsibility to care for these people in the best possible social and economical ways. So, there is a clear interest to create Ambient Intelligence devices and environments allowing the patients to be followed in their own homes or during their day-by-day life.

The medical control support devices may be embedded in clothes, like T-shirts, collecting vital-sign information from sensors (e. g. blood pressure, temperature). Patients will be monitored at long distance. The surrounding environment, for example the patient home, may be aware of the results from the clinical data and even perform emergency calls to order an ambulance service.

For instance, we may refer the IST Vivago® system (IST International Security Technology Oy, Helsinki, Finland), an active social alarm system, which combines intelligent social alarms with continuous remote monitoring of the user's activity profile (Särelä, Korhonen, Lötjönen, Sola, Myllymäki, 2003).

AmI in Tourism and Cultural Heritage

Tourism and Cultural Heritage are good application areas for Ambient Intelligence. Tourism is a growing industry. In the past tourists were satisfied with pre-defined tours, equal for all the people. However there is a trend in the customization and the same tour can be conceived to adapt to tourists according their preferences.

Immersive tour post is an example of such experience (Park, Nam, Shi, Golub, Van Loan, 2006). MEGA is an user-friend virtual-guide to assist visitors in the Parco Archeologico della Valle del Temple in Agrigento, an archaeological area with ancient Greek temples in Agrigento, located in Sicily, Italy (Pilato, Augello, Santangelo, Gentile, Gaglio, 2006). DALICA has been used for constructing and updating the user profile of visitors of Villa Adriana in Tivoli, near Rome, Italy (Constantini, Inverardi, Mostarda, Tocchio, Tsintza, 2007).

AmI at Work

The human being spends considerable time in working places like offices, meeting rooms, manufacturing plants, control centres.

SPARSE is a project initially created for helping Power Systems Control Centre Operators in the diagnosis and restoration of incidents (Vale, Moura, Fernandes, Marques, Rosado, Ramos, 1997). It is a good example of context awareness since the developed system is aware of the on-going situation, acting in different ways according the normal or critical situation of the power system. This system is evolving for an Ambient Intelligence framework applied to Control Centres.

Decision Making is one of the most important activities of the human being. Nowadays decisions imply to consider many different points of view, so decisions are commonly taken by formal or informal groups of persons. Groups exchange ideas or engage in a process of argumentation and counter-argumentation, negotiate, cooperate, collaborate or even discuss techniques and/or methodologies for problem solving. Group Decision Making is a social activity in which the discussion and results consider a combination of rational and emotional aspects. ArgEmotionAgents is a project in the area of the application of Ambient Intelligence in the group argumentation and decision support considering emotional aspects and running in the Laboratory of Ambient Intelligence for Decision Support (LAID), seen in Figure 1 (Marreiros, Santos, Ramos, Neves, Novais, Machado, Bulas-Cruz, 2007), a kind of an Intelligent Decision Room. This work has also a part involving ubiquity support.

AmI in Sports

Sports involve high-level athletes and many more practitioners. Many sports are done without any help of the associated devices, opening here a clear opportunity for Ambient Intelligence to create sports assistance devices and environments.

FlyMaster NAV+ is a free-flight on-board pilot Assistant (e.g. gliding, paragliding), using the FlyMaster F1 module with access to GPS and sensorial information. FlyMaster Avionics S.A., a spin-off, was created to commercialize these products (see figure 2).

AMBIENT INTELLIGENCE PLATFORMS

Some companies and academic institutions are investing in the creation of Ambient Intelligence generation platforms.

The Endeavour project is developed by the California University in Berkeley (http://endeavour.cs.berkeley.edu/). The project aims to specify, design, and implement prototypes at a planet scale, self organized and involving an adaptive "Information Utility".

Oxygen enables pervasive human centred computing through a combination of specific user and system technologies (http://www.oxygen.lcs.mit.edu/). This project provides speech and vision technologies enabling us to communicate with Oxygen as if we were interacting with another person, saving much time and effort (Rudolph, 2001).

The Portolano project was developed in the University of Washington and seeks to create a testbed for research into the emerging field of invisible computing (http://portolano.cs.washington.edu/). The invisible computing is possible with devices so highly optimized to particular tasks that they bend into the world and require little technical knowledge from the users (Esler, Hightower, Anderson, Borrielo, 1999).

The EasyLiving project of Microsoft Research Vision Group corresponds to a prototype architecture and associated technologies for building intelligent environments (Brumitt, Meyers, Krumm, Kern, Shafer,

Figure 1. Ambient Intelligence for decision support, LAID Laboratory

Figure 2. FlyMaster Pilot Assistant device, from FlyMaster Avionics S.A.

2000). EasyLiving goal is to facilitate the interaction of people with other people, with computer, and with devices (http://research.microsoft.com/easyliving/).

FUTURE TRENDS

Ambient Intelligence deals with a futuristic notion for our lives. Most of the practical experiences concerning Ambient Intelligence are still in a very incipient phase, due to the recent existence of this concept. Today, it is not clear the separation between the computer and the environments. However, for new generations things will be more transparent, and environments with Ambient Intelligence will be more widely accepted.

In the area of transport, AmI will cover several aspects. The first will be related with the vehicle itself. Several performances start to be available, like the automatic identification of the situation (e.g. pre-collision identification, identification of the driver conditions). Other aspects will be related with the traffic information. Today, GPS devices are generalized, but they deal with static information. Joining on-line traffic conditions will enable the driver to avoid roads with accidents. Technology is giving good steps in the direction of automatic vehicle driving. But in the near future the developed systems will be seen more like driver assistants in spite of autonomous driving systems.

Another area where AmI will experience a strong development will be the area of Health Care, especially in the Elderly Care. Patients will receive this support to allow a more autonomous life in their homes. However automatic acquisition of vital signals (e.g. blood pressure, temperature) will allow to do automatic emergency calls when the patient health is in significant trouble. The person monitoring will also be done in his/her home, trying to detect differences in expected situations and habits.

The home support will achieve the normal personal and family life. Intelligent Homes will be a reality. The home residents will pay less attention to normal home management aspects, for example, how many bottles of red wine are available for the week meals or if the specific ingredients for a cake are all available.

AmI for job support are also expected. Decision Support Systems will be oriented to on-the-job environments. This will be clear in offices, meeting rooms, call centres, control centres, and plants.

CONCLUSION

This article presents the state of the art in which concerns Ambient Intelligence field. After the history of the concept, we established some related concepts definitions and illustrated with some examples. There is a long way to follow in order to achieve the Ambient Intelligence concept, however in the future, this concept will be referred as one of the landmarks in the Artificial Intelligence development.

REFERENCES

Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S. (2000). EasyLiving: Technologies for Intelligent Environments. Lecture Notes in Computer Science, vol. 1927, pp. 97-119.

Constantini, S., Inverardi, P., Mostarda, L., Tocchio, A., Tsintza, P. (2007). User Profile Agents for Cultural Heritage fruition. Artificial and Ambient Intelligence. Proc. of the Artificial Intelligence and Simulation of Behaviour Annual Convention, pp. 30-33.

Dillon, T. (2006). Pervasive and Ubiquitous Computing. Futurelab. Available at http://www.futurelab.org.uk/viewpoint/art71.htm.

ISTAG (2001), *Scenarios for Ambient Intelligence in 2010*, European Commission Report.

ISTAG (2002). *Strategic Orientations & Priorities for IST in FP6*, European Commission Report.

ISTAG (2003). *Ambient Intelligence: from vision to reality*, European Commission Report.

Marreiros, G., Santos, R., Ramos, C., Neves, J., Novais, P., Machado, J., Bulas-Cruz, J. (2007). Ambient Intelligence in Emotion Based Ubiquitous Decision Making. Proc. Artificial Intelligence Techniques for Ambient Intelligence, IJCAI'07 – Twentieth International Joint Conference on Artificial Intelligence. Hyderabad, India.

McCulloch, W.S., & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*. (5) 115-133.

Park, D., Nam, T., Shi, C., Golub, G., Van Loan, C. (2006). Designing an immersive tour experience system for cultural tour sites. ACM Press. New York, NY. pp. 1193-1198.

Pilato, G., Augello, A., Santangelo, A., Gentile, A., Gaglio S. (2006). An intelligent multimodal site-guide for the Parco Archeologico della Valle del Temple in Agrigento. Proc. of the First Workshop in Intelligent Technologies for Cultural HeritageExploitation. European Conference on Artificial Intelligence.

Esler, M., Hightower, J., Anderson, T., Borrielo, J. (1999). Next century challenges: data-centric networking for invisible computing: the Portolano project at the University of Washington. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp. 256-262.

Ramos, C. (2007). Ambient Intelligence – a State of the Art from Artificial Intelligence perspective. Proceedings of EPIA'2007 – the Portuguese Conference on Artificial Intelligence.

Rothi J., Yen D.(1990). Why American Express Gambled on an Expert Data Base. Information Strategy: The Executive´s Journal, 6(3), pp. 16-22.

Rudolph, L. (2001). Project Oxygen: Pervasive, Human-Centric Computing - An Initial Experience. Lecture Notes in Computer Science, vol. 2068.

Särelä A., Korhonen I., Lötjönen L., Sola M., Myllymäki M. (2003), IST Vivago® - an intelligent social and remote wellness monitoring system for the elderly. In: Proceedings of the 4th Annual IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, pp. 362-365.

Shortliffe, E. (1976). *Computer-Based Medical Consultations: MYCIN*; Elsevier - North Holland.

Thorpe, C., Hebert, M.H., Kanade, T., Shafer, S.A. (1988), Vision and navigation for the Carnegie-Mellon Navlab, IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(3), 362-373.

Vale, Z., Moura, A., Fernandes, M., Marques, A., Rosado, A., Ramos, C. (1997). SPARSE: An Intelligent Alarm Processor and Operator Assistant, IEEE Expert-Special Track on AI Applications in the Electric Power Industry, 12(3), pp. 86- 93, 1997.

Weiser, M. (1991), The Computer for the Twenty-First Century. *Scientific American*. September 1991. pp. 94-104.

Wichert R., Hellenschmidt M. (2006). Intelligent Systems. *Ambient Intelligence solutions for Intelligent Envioronments*. Thematic Brochure of INI-Graphics-Net, pp. 12-13, n.1, 2006.

TERMS AND DEFINITIONS

Ambient Intelligence: Ambient Intelligence (AmI) deals with a new world where computing devices are spread everywhere, allowing the human being to interact in physical world environments in an intelligent and unobtrusive way. These environments should be aware of the needs of people, customizing requirements and forecasting behaviours.

Context Awareness: Context Awareness means that the system is aware of the current situation we are dealing with.

Embedded Systems: Embedded Systems means that electronic and computing devices are embedded in current objects or goods.

Intelligent Decision Room: A decision-making space, eg a meeting room or a control center, equipped with intelligent devices and/or systems to support decision-making processes.

Intelligent Home: A home equipped with several electronic and interactive devices to help residents to manage conventional home decisions.

Intelligent Transportation Systems: Intelligent Systems applied to the area of Transports, namely to traffic and travelling issues.

Intelligent Vehicles: A vehicle equipped with sensors and decision support components.

Pervasive Computing: Pervasive Computing is related with all the physical parts of our lives, the user may have not notion of the computing devices and details related with these physical parts.

Ubiquitous Computing: Ubiquitous Computing means that we have access to computing devices anywhere in an integrated and coherent way.

Analytics for Noisy Unstructured Text Data I

Shourya Roy
IBM Research, India Research Lab, India

L. Venkata Subramaniam
IBM Research, India Research Lab, India

INTRODUCTION

Accdrnig to rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in what oredr the ltteers in a wrod are, the olny iprmoetnt tihng is that the frist and lsat ltteer be at the rghit pclae. Tihs is bcuseae the human mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.[1]

Unfortunately computing systems are not yet as smart as the human mind. Over the last couple of years a significant number of researchers have been focussing on noisy text analytics. Noisy text data is found in informal settings (online chat, SMS, e-mails, message boards, among others) and in text produced through automated speech recognition or optical character recognition systems. Noise can possibly degrade the performance of other information processing algorithms such as classification, clustering, summarization and information extraction. We will identify some of the key research areas for noisy text and give a brief overview of the state of the art. These areas will be, (i) classification of noisy text, (ii) correcting noisy text, (iii) information extraction from noisy text. We will cover the first one in this chapter and the later two in the next chapter.

We define *noise* in text as any kind of difference in the surface form of an electronic text from the intended, correct or original text. We see such *noisy text* everyday in various forms. Each of them has unique characteristics and hence requires special handling. We introduce some such forms of noisy textual data in this section.

Online Noisy Documents: E-mails, chat logs, scrapbook entries, newsgroup postings, threads in discussion fora, blogs, etc., fall under this category. People are typically less careful about the sanity of written content in such informal modes of communication. These are characterized by frequent misspellings, commonly and not so commonly used abbreviations, incomplete sentences, missing punctuations and so on. Almost always noisy documents are human interpretable, if not by everyone, at least by intended readers.

SMS: Short Message Services are becoming more and more common. Language usage over SMS text significantly differs from the standard form of the language. An urge towards shorter message length facilitating faster typing and the need for semantic clarity, shape the structure of this non-standard form known as the *texting language* (Choudhury et. al., 2007).

Text Generated by ASR Devices: ASR is the process of converting a speech signal to a sequence of words. An ASR system takes speech signal such as monologs, discussions between people, telephonic conversations, etc. as input and produces a string a words, typically not demarcated by punctuations as *transcripts*. An ASR system consists of an acoustic model, a language model and a decoding algorithm. The acoustic model is trained on speech data and their corresponding manual transcripts. The language model is trained on a large monolingual corpus. ASR convert audio into text by searching the acoustic model and language model space using the decoding algorithm. Most conversations at contact centers today between agents and customers are recorded. To do any processing of this data to obtain customer intelligence it is necessary to convert the audio into text.

Text Generated by OCR Devices: Optical character recognition, or 'OCR', is a technology that allows digital images of typed or handwritten text to be transferred into an editable text document. It takes the picture of text and translates the text into Unicode or ASCII. . For handwritten optical character recognition, the rate of recognition is 80% to 90% with clean handwriting.

Call Logs in Contact Centers: Today's contact centers (also known as call centers, BPOs, KPOs) produce huge amounts of unstructured data in the form of call logs apart from emails, call transcriptions, SMS, chat

transcripts etc. Agents are expected to summarize an interaction as soon as they are done with it and before picking up the next one. As the agents work under immense time pressure hence the summary logs are very poorly written and sometimes even difficult for human interpretation. Analysis of such call logs are important to identify problem areas, agent performance, evolving problems etc.

In this chapter we will be focussing on automatic classification of noisy text. Automatic text classification refers to segregating documents into different topics depending on content. For example, categorizing customer emails according to topics such as billing problem, address change, product enquiry etc. It has important applications in the field of email categorization, building and maintaining web directories e.g. DMoz, spam filter, automatic call and email routing in contact center, pornographic material filter and so on.

NOISY TEXT CATEGORIZATION

The text classification task is one of the learning models for a given set of classes and applying these models to new unseen documents for class assignment. This is an important component in many knowledge extraction tasks; real time sorting of email or files into folder hierarchies, topic identification to support topic-specific processing operations, structured search and/or browsing, or finding documents corresponding to long-term standing interests or more dynamic task-based interests. Two types of classifiers are generally commonly found viz. *statistical classifiers* and *rule based classifiers*.

In statistical techniques a *model* is typically trained on a corpus of labelled data and once trained the system can be used for automatic assignment of unseen data. A survey of text classification can be found in the work by Aas & Eikvil (Aas & Eikvil, 1999). Given a training document collection $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ with true classes $\{y_1, y_2, \ldots, y_M\}$ the task is to learn a *model*. This model is used for categorizing a new unlabelled document d_u. Typically words appearing in the text are used as features. Other applications including search rely heavily on taking the markup or link structure of documents into account but classifiers only depend on the content of the documents or the collection of words present in the documents. Once features are extracted

from documents, each document is converted into a document vector. Documents are represented in a vector space; each dimension of this space represents a single feature and the importance of that feature in that document gives the exact distance from the origin. The simplest representation of document vectors uses the binary event model, where if a feature $j \in V$ appears in document d_i, then the jth component of d_i is 1 otherwise it is 0. One of the most popular statistical classification techniques is naive Bayes (McCallum, 1998). In the naive Bayes technique the probability of a document d_i belonging to class c is computed as:

$$\Pr(c \mid d) = \frac{\Pr(c, d)}{\Pr(d)}$$

$$= \frac{\Pr(c) \Pr(d \mid c)}{\Pr(d)}$$

$$\infty \ \Pr(c) \Pr(d \mid c)$$

$$\infty \ \prod_{j} P(d_j \mid c)$$

The final approximation of the above equation refers to the naive part of such a model, i.e., the assumption of word independence which means the features are assumed to be conditionally independent, given the class variable.

Rule-based learning systems have been adopted in the document classification problem since it has considerable appeal. They perform well at finding simple axis-parallel frontiers. A typical rule-based classification scheme for a category, say *C*, has the form:

Assign category C if antecedent or
Do no assign category C if antecedent or

The antecedent in the premise of a rule usually involves some kind of feature value comparison. A rule is said to cover a document or a document is said to satisfy a rule if all the feature value comparisons in the antecedent of the rule are true for the document. One of the well known works in the rule based text classification domain is RIPPER. Like a standard separate-and-conquer algorithm, it builds a rule set incrementally. When a rule is found, all documents covered by the rule are discarded including positive

and negative documents. The rule is then added to the rule set. The remaining documents are used to build other rules in the next iteration.

In both statistical as well as rule based text classification techniques, the content of the text is the sole determiner of the category to be assigned. However noise in the text distorts the content and hence readers can expect the categorization performance to get affected by noise in the text. Classifiers are essentially trained to identify correlation between extracted features (words) with different categories which can be later utilized to categorize new documents. For example, words like *exciting offer get a free laptop* might have stronger correlation with category *spam emails* than non-spam emails. Noise in text distorts this feature space *excitinng ofer get frree lap top* will be new set of features and the categorizer will not be able to relate it to the *spam emails* category. The feature space explodes as the same feature can appear in different forms due to spelling errors, poor recognition, wrong transcription, etc. In the remaining part of this section we will give an overview how people have approached the problem of categorizing noisy text.

Categorization of OCRed Documents

Electronically recognized handwritten documents and documents generated from OCR process are typical examples of noisy text because of the errors introduced by the recognition process. Vinciarelli (Vinciarelli, 2004) has studied the characteristics of noise present in such data and its effects on categorization accuracy. A subset of documents from the Reuters-21578 text classification dataset were taken and noise was introduced using two methods: first a subset of documents were manually written and recognized using an offline handwriting recognition system. In the second the OCR based extraction process was simulated by randomly changing a certain percentage of characters. According to them for recall values up to 60-70 percent depending on the sources, the categorization system is robust to noise even when the Term Error Rate is higher than 40 percent. It was also observed that the results from the handwritten data appeared to be lower than those obtained from OCR simulations. Generic systems for text categorization based on statistical analysis of representative text corpora have been proposed (Bayer et. al., 1998). Features are extracted from training texts by selecting substrings from actual word forms and

applying statistical information and general linguistic knowledge followed by dimensionality reduction by linear transformation. The actual categorization system is based on minimum least-squares approach. The system is evaluated on the tasks of categorizing abstracts of paper-based German technical reports and business letters concerning complaints. Approximately 80% classification accuracy is obtained and it is seen that the system is very robust against recognition or typing errors.

Issues with categorizing OCRed documents are also discussed by many other authors (Brooks & Teahan, 2007), (Hoch, 1994) and (Taghva et. al., 2001).

Categorization of ASRed Documents

Automatic Speech Recognition (ASR) is simply the process of converting an acoustic signal to a sequence of words. Researchers have proposed different techniques for speech recognition tasks based on Hidden Markov model (HMM), neural networks, Dynamic time warping (DTW) (Trentin & Gori, 2001). The performance of an ASR system is typically measured in terms of Word Error Rate (WER), which is derived from the Levenshtein distance, working at word level instead of character. WER can be computed as

$$WER = \frac{S + D + I}{N}$$

where S is the number of substitutions, D is the number of the deletions, I is the number of the insertions, and N is the number of words in the reference. Bahl et.al. (Bahl et. al. 1995) have built an ASR system and demonstrated its capability on benchmark datasets.

ASR systems give rise to word substitutions, deletions and insertions, while OCR systems produce essentially word substitutions. Moreover, ASR systems are constrained by a lexicon and can give as output only words belonging to it, while OCR systems can work without a lexicon (this corresponds to the possibility of transcribing any character string) and can output sequences of symbols not necessarily corresponding to actual words. Such differences are expected to have strong influence on performance of systems designed for categorizing ASRed documents in comparison to categorization of OCRed documents. A lot of work on automatic call type classification for the purpose of

categorizing calls (Tang et al., 2003), call routing (Kuo and Lee, 2003; Haffner et al., 2003), obtaining call log summaries (Douglas et al., 2005), agent assisting and monitoring (Mishne et al., 2005) has appeared in the past. Here calls are classified based on the transcription from an ASR system. One interesting work on seeing effect of ASR noise on text classification was done on a subset of benchmark text classification dataset Reuters-21578[2] (Agarwal et. al., 2007). They read out and automatically transcribed 200 documents and applied a text classifier trained on clean Reuters-21578 training corpus[3]. Surprisingly, in spite of high degree of noise, they did not observe much degradation in accuracy.

Effect of Spelling Errors on Categorization

Spelling errors are an integral part of written text—electronic as well as non-electronic. Every reader reading this book must have been scolded by their teacher in school for spelling words wrongly! In this era of electronic text people have become less careful while writing resulting poorly written text containing abbreviations, short forms, acronyms, wrong spellings. Such electronic text documents including email, chat log, postings, SMSs are sometimes difficult to interpret even for human beings. It goes without saying that text analytics on such noisy data is a non trivial task.

Wrong spellings can affect automatic classification performance in multiple ways depending on the nature of the classification technique being used. In the case of statistical techniques, spelling differences distort the feature space. If training as well as the test data corpus are noisy, while learning the model the classifier will treat variants of the same words as different features. As a result the observed joint probability distribution will be different from the actual distribution. If the proportion of wrongly spelt words is high then the distortion can be significant and will hurt the accuracy of the resultant classifier. However, if the classifier is trained on a clean corpus and the test documents are noisy, then wrongly spelt words will be treated as unseen words and will not help in classification. In an unlikely situation a wrongly spelt word present in a test document may become a different valid feature and worse, may become a valid indicative feature of a different class. A standard technique in the text classification process is *feature selection* which happens after *feature extraction* and before *training*. Feature

selection typically employs some statistical measures over the training corpus and ranks features in order of the amount of information (correlation) they have with respect to the class labels of the classification task at hand. After the feature set has been ranked, the top few features are retained (typically order of hundreds or a few thousand) and the others are discarded. Feature selection should be able to eliminate wrongly spelt words present in the training data provided (i) the proportion of wrongly spelt words is not very large and (ii) there is no regular pattern in spelling errors[4]. However it has been observed, even at high degree of spelling errors the classification accuracy does not suffer much (Agarwal et al., 2007).

Rule based classification techniques also get negatively affected by spelling errors. If the training data contains spelling errors then some of the rules may not get the required statistical significance. Due to spelling errors present in the test data a valid rule may not fire and worse, an invalid rule may fire leading to a wrong categorization. Suppose RIPPER has learnt a rule set like:

Assign category "sports" IF
(the document contains {\it sports}) OR
(the document contains {\it exercise} AND {\it outdoor}) OR
(the document contains {\it exercise} but not {\it homework} {\it exam}) OR
(the document contains {\it play} AND {\it rule}) OR
……

A hypothetical test document containing repeated occurrences of *exercise,* but each time wrongly spelt as *exarcise,* will not be categorized to the *sports* category and hence lead to misclassification.

CONCLUSION

In this chapter we have looked at noisy text analytics. This topic is gaining in importance as more and more noisy data gets generated and needs processing. In particular we have looked at techniques for correcting noisy text and for doing classification. We have presented a survey of existing techniques in the area and have shown that even though it is a difficult problem it is possible to address it with a combination of new and existing techniques.

A

REFERENCES

K. Aas & L. Eikvil (1999). Text Categorisation: A Survey. Technical report, Norwegian Computing Center.

S. Agarwal, S. Godbole, D. Punjani & S. Roy (2007). How Much Noise is too Much: A Study in Automatic Text Classification. In Proceedings of the IEEE International Conference on Data Mining series (ICDM), Nebraska, Omaha (To Appear).

L. R. Bahl, S. Balakrishnan-Aiyer, J. Bellegarda, M. Franz, P. Gopalakrishnan, D. Nahamoo, M. Novak, M. Padmanabhan, M. Picheny, and S. Roukos. Performance of the IBM large vocabulary continuous speech recognition system on the ARPA wall street journal task. In Proc. ICASSP '95, pages 41–44, Detroit, MI, 1995.

T. Bayer, U. Kressel, H. Mogg-Schneider, & Renz (1998). Categorizing Paper Documents. Computer Vision and Image Understanding, 70(3) (299-306).

R. Brooks & L. J. Teahan (2007). A Practical Implementation of Automatic Text Categorization and Correction of the Conversion of Noisy OCR Documents into Braille and Large Print. Proceedings of Workshop on Analytics for Noisy Unstructured Text Data (at IJCAI 2007). Jan, Hyderabad, India.

S. Douglas, D. Agarwal, T. Alonso, R. M. Bell, M. Gilbert, D. F. Swayne and C. Volinsky. 2005. Mining Customer Care Dialogs for "Daily News". IEEE Trans. on Speech and Audio Processing, 13(5):652–660.

P. Haffner, G. Tur & J. H. Wright (2003). Optimizing SVMs for Complex Call Classification. In Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing.

R. Hoch (1994). Using IR Techniques for Text Classification in Document Analysis. In Proceedings of 17th ACM SIGIR Conference on Research and Development in Information Retrieval, (31-40).

H.-K J. Kuo and C.-H. Lee. 2003. Discriminative Training of Natural Language Call Routers. IEEE Trans. on Speech and Audio Processing, 11(1):24–35.

A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In AAAI/ICML-98 Workshop on Learning for Text Categorization, 1998.

G. Mishne, D. Carmel, R. Hoory, A. Roytman and A. Soffer. 2005. Automatic Analysis of Call-center Conversations. Conference on Information and Knowledge Management. October 31-November 5, Bremen, Germany.

K. Taghva, T. Narkter, J. Borsack, Lumos. S., A. Condit, & Young (2001). Evaluating Text Categorization in the Presence of OCR Errors. In Proceedings of IS&T SPIE 2001 International Symposium on Electronic Imaging Science and Technology, (68-74).

M. Tang, B. Pellom and K. Hacioglu. 2003. Calltype Classification and Unsupervised Training for the Call Center Domain. Automatic Speech Recognition and Understanding Workshop. November 30-December 4, St. Thomas, U S Virgin Islands.

E. Trentin & M. Gori (2001). A Survey of Hybrid ANN/HMM Models for Automatic Speech Recognition. Neurocomputing journal. Volume 37. (91-126)

A. Vinciarelli (2005). Noisy Text Categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, no. 12. (1882 – 1295).

Vlachos (2006). Active Annotation. In *Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining,* Trento, Italy.

KEY TERMS

Automatic Speech Recognition: Machine recognition and conversion of spoken words into text.

Data Mining: The application of analytical methods and tools to data for the purpose of identifying patterns, relationships or obtaining systems that perform useful tasks such as classification, prediction, estimation, or affinity grouping.

Information Extraction: Automatic extraction of structured knowledge from unstructured documents.

Noisy Text: Text with any kind of difference in the surface form, from the intended, correct or original text.

Optical Character Recognition: Translation of images of handwritten or typewritten text (usually captured by a scanner) into machine-editable text.

Rule Induction: Process of learning, from cases or instances, if-then rule relationships that consist of an antecedent (if-part, defining the preconditions or coverage of the rule) and a consequent (then-part, stating a classification, prediction, or other expression of a property that holds for cases defined in the antecedent).

Text Analytics: The process of extracting useful and structured knowledge from unstructured documents to find useful associations and insights.

Text Classification (or Text Categorization): Is the task of learning models for a given set of classes and applying these models to new unseen documents for class assignment.

ENDNOTES

[1] According to http://www.mrc-cbu.cam.ac.uk/%7Emattd/Cmabrigde/, this is an internet hoax. However we found it interesting and hence included here.

[2] http://www.daviddlewis.com/resources/testcollections/

[3] This dataset is available from http://kdd.ics.uci.edu/databases/reuters_transcribed/reuters_transcribed.html

[4] Note: this assumption may not hold true in the case of cognitive errors

Analytics for Noisy Unstructured Text Data II

L. Venkata Subramaniam
IBM Research, India Research Lab, India

Shourya Roy
IBM Research, India Research Lab, India

INTRODUCTION

The importance of text mining applications is growing proportionally with the exponential growth of electronic text. Along with the growth of internet many other sources of electronic text have become really popular. With increasing penetration of internet, many forms of communication and interaction such as email, chat, newsgroups, blogs, discussion groups, scraps etc. have become increasingly popular. These generate huge amount of noisy text data everyday. Apart from these the other big contributors in the pool of electronic text documents are call centres and customer relationship management organizations in the form of call logs, call transcriptions, problem tickets, complaint emails etc., electronic text generated by Optical Character Recognition (OCR) process from hand written and printed documents and mobile text such as Short Message Service (SMS). Though the nature of each of these documents is different but there is a common thread between all of these—presence of noise.

An example of information extraction is the extraction of instances of corporate mergers, more formally *MergerBetween(company1,company2,date)*, from an online news sentence such as: *"Yesterday, New-York based Foo Inc. announced their acquisition of Bar Corp." Opinion(product1,good)*, from a blog post such as: *"I absolutely liked the texture of SheetK quilts."*

At superficial level, there are two ways for information extraction from noisy text. The first one is cleaning text by removing noise and then applying existing state of the art techniques for information extraction. There in lies the importance of techniques for automatically correcting noisy text. In this chapter, first we will review some work in the area of noisy text correction. The second approach is to devise extraction techniques which are robust with respect to noise. Later in this chapter, we will see how the task of information extraction is affected by noise.

NOISY TEXT CORRECTION

Before moving on to techniques for processing noisy text we will briefly introduce methods for correcting noisy text. One of the most common forms of noise in text is wrong spelling. Kukich provides a comprehensive survey of techniques pertaining to detecting and correcting spelling errors (Kukich, 1992). According to this survey, three types of nonword misspellings are typically found viz. **typographic** such as *teh*, *speel*, **cognitive** such as *recieve*, *conspeeracy* and **phonetic** such as *abiss*, *nacherly*. A distinction must be made between automatically *detecting* such errors and automatically *correcting* those errors. The latter is a much harder problem. Most of the recent work in this area is about correcting spelling mistakes automatically. Golding and Roth (Golding & Roth, 1999) proposed a combination of a variant of *Winnow*, a multiplicative weight-update algorithm and weighted majority voting for context sensitive spelling correction. Mangu and Brill (Mangu & Brill, 1997) have shown that a small set of human understandable rules is more meaningful than a large set of opaque features and weights. Hybrid methods capturing the context using trigrams of the parts-of-speech tags and a feature based method have also been proposed to handle context sensitive spelling correction (Golding & Schabes, 1996). There is a lot of work related to automatic correction of spelling errors (Agirre et. al., 1998), (Zamora et. al., 1983), (Golding, 1995). A complete bibliography of all the work related to spelling error detection and correction can be found in (Beebe, 2005). On a related note, automatic spelling error correction techniques have been applied for other

applications such as semantic role labelling (Sang et. al., 2005).

There is also recent work on correcting the output of SMS text (Aw et. al., 2006) (Choudhury et. al., 2007), OCR errors (Nartker et. al., 2003) and ASR errors (Sarma & Palmer, 2004).

INFORMATION EXTRACTION FROM NOISY TEXT

The goal of Information Extraction (IE) is to automatically extract structured information from the unstructured documents. The extracted structured information has to be contextually and semantically well-defined data from a given domain. A typical application of IE is to scan a set of documents written in natural language and populate a database with the information extracted. The MUC (Message Understanding Conference) conference was one effort at codifying the IE task and expanding it (Chinchor, 1998).

There are two basic approaches to the design of IE systems. One comprises the *knowledge engineering approach* where a domain expert writes a set of rules to extract the sought after information. Typically the process of building the system is iterative whereby a set of rules is written, the system is run and the output examined to see how the system is performing. The domain expert then modifies the rules to overcome any under- or over-generation in the output. The second is the *automatic training approach*. This approach is similar to classification where the texts are appropriately annotated with the information being extracted. For example, if we would like to build a city name extractor, then the training set would include documents with all the city names marked. An IE system would be trained on this annotated corpus to learn the patterns that would help in extracting the necessary entities.

An information extraction system typically consists of natural language processing steps such as morphological processing, lexical processing and syntactic analysis. These include stemming to reduce inflected forms of words to their stem, parts of speech tagging to assign labels such as noun, verb, etc. to each word and parsing to determine the grammatical structure of sentences.

Named Entity Annotation of Web Posts

Extraction of named entities is a key IE task. It seeks to locate and classify atomic elements in the text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. Entity recognition systems either use rule based techniques or statistical models. Typically a parser or a parts of speech tagger identifies elements such as nouns, noun phrases, or pronouns. These elements along with surface forms of the text are used to define templates for extracting the named entities. For example, to tag company names it would be desirable to look at noun phrases that contain the words *company* or *incorporated* in them. These rules can be automatically learnt using a tagged corpus or could be defined manually. Most known approaches do this on clean well formed text. However, named entity annotation of web posts such as online classifieds, product listings etc. is harder because these texts are not grammatical or well written. In such cases reference sets have been used to annotate parts of the posts (Michelson & Knoblock, 2005). The reference set is thought of as a relational set of data with a defined schema and consistent attribute values. Posts are now matched to their nearest records in the reference set. In the biological domain gene name annotation, even though it is performed on well written scientific articles, can be thought of in the context of noise, because many gene names overlap with common English words or biomedical terms. There have been studies on the performance of the gene name annotator when trained on noisy data (Vlachos, 2006).

Information Extraction from OCRed Documents

Documents obtained from OCR may have not only unknown words and compound words, but also incorrect words due to OCR errors. In their work Miller et. al. (Miller et. al., 2000) have measured the effect of OCR noise on IE performance. Many IE methods work directly on the document image to avoid errors resulting from converting to text. They adopt keyword matching by searching for string patterns and then use global document models consisting of keyword models and their logical relationships to achieve robustness in matching (Lu & Tan, 2004). The presence of OCR errors has a detrimental effect on information access

from these documents (Taghva et. al., 2004). However, post processing of these documents to correct these errors exist and have been shown to give large improvements.

Information Extraction from ASRed Documents

The output of an ASR system does not contain case information and punctuations. It has been shown that in the absence of punctuations extraction of different syntactic entities like parts of speech and noun phrases is not accurate (Nasukawa et. al., 2007). So IE from ASRed documents becomes harder. Miller et. al. (Miller et. al., 2000) have shown how IE performance varies with ASR noise. It has been shown that it is possible to build aggregate models from ASR data (Roy & Subramaniam, 2006). In this work topical models are constructed by utilizing inter document redundancy to overcome the noise. In this work only a few natural language processing steps have been used. Phrases have been aggregated over the noisy collection to get to the clean underlying text.

FUTURE TRENDS

More and more data from sources like chat, conversations, blogs, discussion groups need to be mined to capture opinions, trends, issues and opportunities. These forms of communication encourage informal language which can be considered noisy due to spelling errors, grammatical errors and informal writing styles. Companies are interested in mining such data to observe customer preferences and improve customer satisfaction. Online agents need to be able to understand web posts to take actions and communicate with other agents. Customers are interested in collated product reviews from web posts of other users. The nature of the noisy text warrants moving beyond traditional text analytics techniques. There is need for developing natural language processing techniques that are robust to noise. Also techniques that implicitly and explicitly tackle textual noise need to be developed.

CONCLUSION

In this chapter we have looked at information extraction from noisy text. This topic is gaining in importance as more and more noisy data gets generated and useful information needs to be obtained from this. We have presented a survey of existing techniques information extraction techniques. We have also presented some of the future trends in noisy text analytics.

REFERENCES

E. Agirre, K. Gojenola, K. Sarasola & A. Voutilainen (1998). Towards a Single Proposal in Spelling Correction. Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics (22-28).

Aw, M. Zhang, J. Xiao & J. Su (2006). A Phrase-Based Statistical Model for SMS Text Normalization. In Proceedings of the Joint conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics (ACL-COLING 2006), Sydney, Australia.

N. H. F. Beebe (2005). A Bibliography of Publications on Computer Based Spelling Error Detection and Correction. http://www.math.utah.edu/pub/tex/bib/spell.ps.gz.

M. Choudhury, R. Saraf, V. Jain, S. Sarkar & A. Basu (2007). Investigation and Modeling of the Structure of Texting Language. In Proceedings of the IJCAI 2007 Workshop on Analytics for Noisy Unstructured Text Data (AND 2007), Hyderabad, India.

N. Chinchor (1998). Overview of MUC-7. http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/overview.html

R. Golding (1995). A Bayesian Hybrid Method for Context-Sensitive Spelling Correction. Proceedings of the Third Workshop on Very Large Corpora (39—53).

R. Golding & D. Roth (1999). A Winnow-Based Approach to Context-Sensitive Spelling Correction. Journal of Machine Learning. Volume 34 (1-3) (107-130)

A

R. Golding & Y. Schabes (1996). Combining Tri-gram-Based and Feature-Based Methods for Context-Sensitive Spelling Correction. Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics (71—78).

K. Kukich (1992). Technique for Automatically Correcting Words in Text. ACM Computing Survey. Volume 24 (4) (377—439).

Y. Lu & C. L. Tan (2004). Information Retrieval in Document Image Databases. IEEE Transactions on Knowledge and Data Engineering. Vol 16, No. 11. (1398-1410)

L. Mangu & E. Brill (1997). Automatic Rule Acquisition for Spelling Correction. Proc. 14th International Conference on Machine Learning. (187—194).

M. Michelson & C. A. Knoblock (2005). Semantic Annotation of Unstructured and Ungrammatical Text. *In Proceedings of the International Joint Conference on Artificial Intelligence*.

D. Miller, S. Boisen, R. Schwartz, R. Stone & R. Wei-schedel (2000). Named Entity Extraction from Noisy Input: Speech and OCR. Proceedings of the Sixth Conference on Applied Natural Language Processing.

T. Nartker, K. Taghva, R. Young, J. Borsack, and A. Condit (2003). OCR Correction Based On Document Level Knowledge. In Proc. IS&T/SPIE 2003 Intl. Symp. on Electronic Imaging Science and Technology, volume 5010, Santa Clara, CA.

T. Nasukawa, D. Punjani, S. Roy, L. V. Subramaniam & H. Takeuchi (2007). Adding Sentence Boundaries to Conversational Speech Transcriptions Using Noisily Labeled Examples. In Proceedings of the IJCAI 2007 Workshop on Analytics for Noisy Unstructured Text Data (AND 2007), Hyderabad, India.

S. Roy & L. V. Subramaniam (2006). Automatic Generation of Domain Models for Call-Centers from Noisy Transcriptions. In Proceedings of the Joint conference of the Association for Computational Linguistics and the International Committee on Computational Linguistics (ACL-COLING 2006), Sydney, Australia.

E. T. K. Sang, S. Canisius, A. van den Bosch & T. Bogers (2005). Applying Spelling Error Correction Techniques for Improving Semantic Role Labelling. In Proceedings of CoNLL.

Sarma & D. Palmer (2004). Context-based Speech Recognition Error Detection and Correction. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004.

K. Taghva, T. Narkter & J. Borsack (2004). Information Access in the Presence of OCR Errors. ACM Hardcopy Document Processing Workshop, Washington, DC, USA. (1-8)

K. Taghva, T. Narkter, J. Borsack, Lumos. S., A. Condit, & Young (2001). Evaluating Text Categorization in the Presence of OCR Errors. In Proceedings of IS&T SPIE 2001 International Symposium on Electronic Imaging Science and Technology, (68-74).

E. M. Zamora, J. J. Pollock, & A. Zamora (1983). The Use of Trigram Analysis for Spelling Error Detection. Information Processing and Management 17. 305-316.

KEY TERMS

Automatic Speech Recognition: Machine recognition and conversion of spoken words into text.

Data Mining: The application of analytical methods and tools to data for the purpose of identifying patterns, relationships or obtaining systems that perform useful tasks such as classification, prediction, estimation, or affinity grouping.

Information Extraction: Automatic extraction of structured knowledge from unstructured documents.

Knowledge Extraction: Explicitation of the internal knowledge of a system or set of data in a way that is easily interpretable by the user.

Noisy Text: Text with any kind of difference in the surface form, from the intended, correct or original text.

Optical Character Recognition: Translation of images of handwritten or typewritten text (usually captured by a scanner) into machine-editable text.

Rule Induction: Process of learning, from cases or instances, if-then rule relationships that consist of an antecedent (if-part, defining the preconditions or

A

coverage of the rule) and a consequent (then-part, stating a classification, prediction, or other expression of a property that holds for cases defined in the antecedent).

Text Analytics: The process of extracting useful and structured knowledge from unstructured documents to find useful associations and insights.

Angiographic Images Segmentation Techniques

Francisco J. Nóvoa
University of A Coruña, Spain

Alberto Curra
University of A Coruña, Spain

M. Gloria López
University of A Coruña, Spain

Virginia Mato
University of A Coruña, Spain

INTRODUCTION

Heart-related pathologies are among the most frequent health problems in western society. Symptoms that point towards cardiovascular diseases are usually diagnosed with **angiographies**, which allow the medical expert to observe the bloodflow in the coronary arteries and detect severe narrowing (**stenosis**). According to the severity, extension, and location of these narrowings, the expert pronounces a diagnosis, defines a treatment, and establishes a prognosis.

The current modus operandi is for clinical experts to observe the image sequences and take decisions on the basis of their empirical knowledge. Various techniques and **segmentation** strategies now aim at objectivizing this process by extracting quantitative and qualitative information from the **angiographies**.

BACKGROUND

Segmentation is the process that divides an image in its constituting parts or objects. In the present context, it consists in separating the pixels that compose the coronary tree from the remaining "background" pixels.

None of the currently applied **segmentation** methods is able to completely and perfectly extract the vasculature of the heart, because the images present complex morphologies and their background is inhomogeneous due to the presence of other anatomic elements and artifacts such as catheters.

The literature presents a wide array of coronary tree extraction methods: some apply pattern recognition techniques based on pure intensity, such as **thresholding** followed by an analysis of connected components, whereas others apply explicit vessel models to extract the vessel contours.

Depending on the quality and noise of the image, some **segmentation** methods may require image preprocessing prior to the **segmentation** algorithm; others may need postprocessing operations to eliminate the effects of a possible oversegmentation.

The techniques and algorithms for vascular **segmentation** could be categorized as follows (Kirbas, Quek, 2004):

1. Techniques for "pattern-matching" or pattern recognition
2. Techniques based on models
3. Techniques based on tracking
4. Techniques based on artificial intelligence
5. Main Focus

This section describes the main features of the most commonly accepted coronary tree **segmentation** techniques. These techniques automatically detect objects and their characteristics, which is an easy and immediate task for humans, but an extremely complex process for artificial computational systems.

Techniques Based on Pattern Recognition

The pattern recognition approaches can be classified into four major categories:

Figure 1. Regions growth applied to an angiography
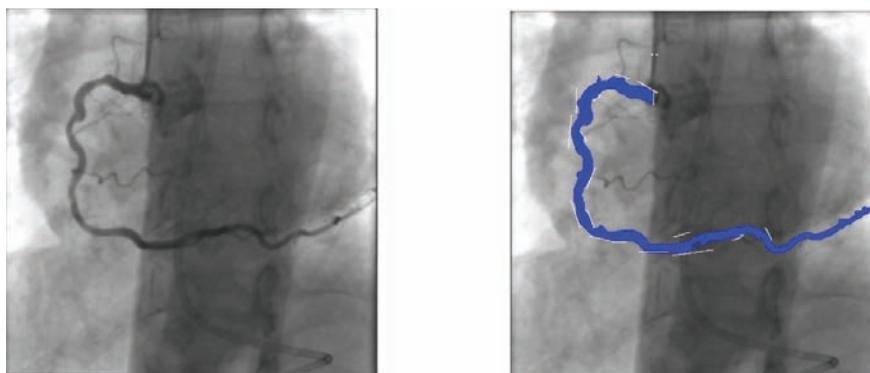
Multiscale Methods

The multiscale method extracts the vessel method by means of images of varying resolutions. The main advantage of this technique resides in its high speed. Larger structures such as main arteries are extracted by segmenting low resolution images, whereas smaller structures are obtained through high resolution images.

Methods Based on Skeletons

The purpose of these methods is to obtain a *skeleton* of the coronary tree: a structure of smaller dimensions than the original that preserves the topological properties and the general shape of the detected object. Skeletons based on curves are generally used to reconstruct vascular structures (Nyström, Sanniti di Baja & Svensson, 2001). Skeletonizing algorithms are also called "thinning algorithms".

The first step of the process is to detect the central axis of the vessels or "centerline". This axis is an imaginary line that follows each vessel in its central axis, i.e. two normal segments that cross the axis in opposite sense should present the same distance from the vessel's edges. The total of these lines constitutes the skeleton of the coronary tree. The methods that are used to detect the central axes can be classified into three categories:

Methods Based on Crests
One of the first methods to segment angiographic images on the basis of crests was proposed by Guo and

Richardson (Guo & Ritchardson, 1998). This method treats **angiographies** as topographic maps in which the detected crests constitute the central axes of the vessels.

The image is preprocessed by means of a median filter and smoothened with non-linear diffusion. The region of interest is then selected through **thresholding**, a process that eliminates the crests that do not correspond with the central axes. Finally, the candidate central axes are joined with curve relaxation techniques.

Methods Based on Regions Growth
Taking a known point as seed point, these techniques segment images through the incremental inclusion of pixels in a region on the basis of an *a priori* established criterion. There are two especially important criteria: similitude in the value, and spatial proximity (Jain, Kasturi & Schunck, 1995). It is established that pixels that are sufficiently near others with similar grey levels belong to the same object. The main disadvantage of this method is that it requires the intervention of the user to determine the seed points.

O'Brien and Ezquerra (O'Brien & Ezquerra, 1994) propose the automatic extraction of the coronary vessels in angiograms on the basis of temporary, spatial, and structural restrictions. The algorithm starts with a low-pass filter and the user's definition of a seed point. The system then starts to extract the central axes by means of the "globe test" mechanism, after which the detected regions are entangled through the graph theory. The applied test also allows us to discard the regions that are detected incorrectly and do not belong to the vascular tree.

111

Methods Based on Differential Geometry

The methods that are based on differential geometry treat images as hypersurfaces and extract their features using curvature and surface crests. The points of hypersurface's crest correspond to the central axis of the structure of a vessel. This method can be applied to bidimensional as well as tridimensional images; angiograms are bidimensional images and are therefore modelled as tridimensional hypersurfaces.

Examples of reconstructions can be found in Prinet et al (Prinet, Mona & Rocchisani, 1995), who treat the images as parametric surfaces and extract their features by means of surfaces and crests.

Correspondence Filters Methods

The correspondence filter approach convolutes the image with multiple correspondence filters so as to extract the regions of interest. The filters are designed to detect different sizes and orientations.

Poli and Valli (Poli, R & Valli, 1997) apply this technique with an algorithm that details a series of multiorientation linear filters that are obtained as linear combinations of Gaussian "kernels". These filters are sensitive to different vessel widths and orientations.

Mao et al (Mao, Ruan, Bruno, Toumoulin, Collorec & Haigron, 1992) also use this type of filters in an algorithm based on visual perception models that affirm that the relevant parts of the objects in images with noise appear normally grouped.

Morphological Mathematical Methods

Mathematical morphology defines a series of operators that apply structural elements to the images so that

their morphological features can be preserved and irrelevant elements eliminated. The main morphological operations are the following:

- **Dilatation:** Expands objects, fills up empty spaces, and connects disjunct regions.
- **Erosion:** Contracts objects, separates regions.
- **Closure:** Dilatation + Erosion.
- **Opening:** Erosion + Dilatation.
- **"Top hat" transformation:** Extracts the structures with a linear shape
- **"Watershed" transformation:** "Inundates" the image that is taken as a topographic map , and extracts the parts that are not "flooded".

Eiho and Qian (Eiho & Qian, 1997) use a purely morphological approach to define an algorithm that consists of the following steps:

1. Application of the "top hat" operator to emphasize the vessels
2. Erosion to eliminate the areas that do not correspond to vessels
3. Extraction of the tree from a point provided by the user and on the basis of grey levels.
4. Slimming down of the tree
5. Extraction of edges through "watershed" transformation

MODEL-BASED TECHNIQUES

These approaches use explicit vessel models to extract the vascular tree. They can be divided into four catego-

Figure 2. Morphological operators applied to an angiography
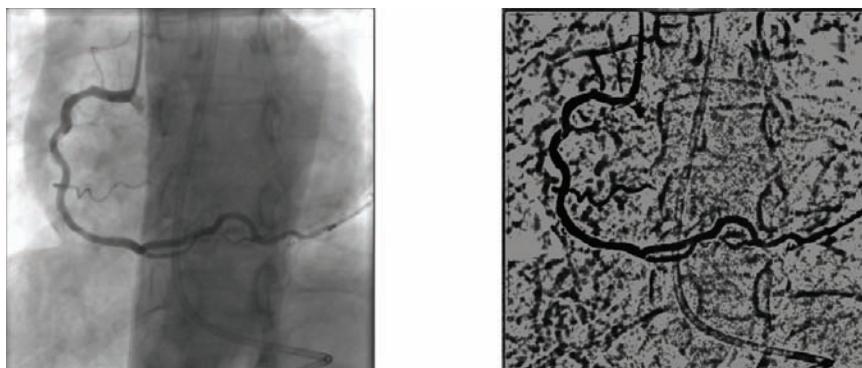
ries: deformable models, parametric models, template correspondence models, and generalized cylinders.

Deformable Models

Strategies based on deformable models can be classified in terms of the work by McInerney and Terzopoulos (McInerney & Terzopoulos, 1997).

Algorithms that use deformable models (Merle, Finet, Lienard, & Magnin, 1997) are based on the progressive refining of an initial skeleton built with curves from a series of reference points:

- **Root points:** Starting points for the coronary tree.
- **Bifurcation points:** Points where a main branch divides into a secundary branch.
- **End points:** Points where a tree branch ends.

These points have to be marked manually.

Deformable Parametric Models: Active Contours

These models use a set of parametric curves that adjust to the object's edges and are modified by both external forces, that foment deformation, and internal forces that resist change. The active contour models or "snakes" in particular are a special case of a more general technique that pretends to adjust deformable models by minimizing energy.

Klein et al. (Klein, Lee & Amini, 1997) propose an algorithm that uses "snakes" for 4D reconstruction: they trace the position of each point of the central axis of a skeleton in a sequence of angiograms.

Deformable Geometric Models

These models are based on topographic models that are adapted for shape recognition. Malladi et al. (Malladi, Sethian & Vemuri, 1995) for instance adapt the "Level Set Method" (LSM) by representing an edge as a level zero set of a hypersurface of a superior order; the model evolves to reduce a metric defined by the restrictions of edges and curvature, but less rigidly than in the case of the "snakes". This edge, which constitutes the zero level of the hypersurface, evolves by adjusting to the edges of the vessels, which is what we want to detect.

Propagation Methods

Quek and Kirbas (Quek & Kirbas, 2001) developed a system of wave propagation combined with a backtracking mechanism to extract the vessels from angiographic images. This method basically labels each pixel according to its likeliness to belong to a vessel and then propagates a wave through the pixels that are labeled as belonging to the vessel; it is this wave that definitively extracts the vessels according to the local features it encounters.

Approaches based on the correspondence of deformable templates:

This approach tries to recognize structural models (templates) in an image by using a template as context, i.e. as *a priori* model. This template is generally represented as a set of nodes connected by a segment. The initial structure is deformed until it adjusts optimally to the structures that were observed in the image.

Petrocelli et al. (Petrocelli, Manbeck, & Elion, 1993) describe a method based on deformable templates that also incorporates additional previous knowledge into the deformation process.

Parametric Models

These models are based on the a priori knowledge of the **artery**'s shape and are used to build models whose parameters depend on the profiles of the entire vessel; as such, they consider the global information of the **artery** instead of merely the local information. The value of these parameters is established after a learning process.

The literature shows the use of models with circular sections (Shmueli, Brody, & Macovski, 1983) and spiral sections (Pappas, & Lim, 1984), because various studies by Brown, B. G., (Bolson, Frimer, & Dodge, 1977) (Brown, Bolson, Frimer & Dodge, 1982) show that sections of healthy arteries tend to be circular and sections with **stenosis** are usually elliptical. However, both circular and elliptical shapes fail to approach irregular shapes caused by pathologies or bifurcations.

This model has been applied to the reconstruction of vascular structures with two angiograms (Pellot, Herment, Sigelle, Horain, Maitre & Peronneau, 1994), which is why both healthy and stenotic sections are modeled by means of ellipses. This model is subsequently deformed until it corresponds to the shape associated to the birth of a new branch or pathology.

A

Figure 3. "Snakes" applied to a blood vessel. *http://vislab.cs.vt.edu/review/extraction.html*
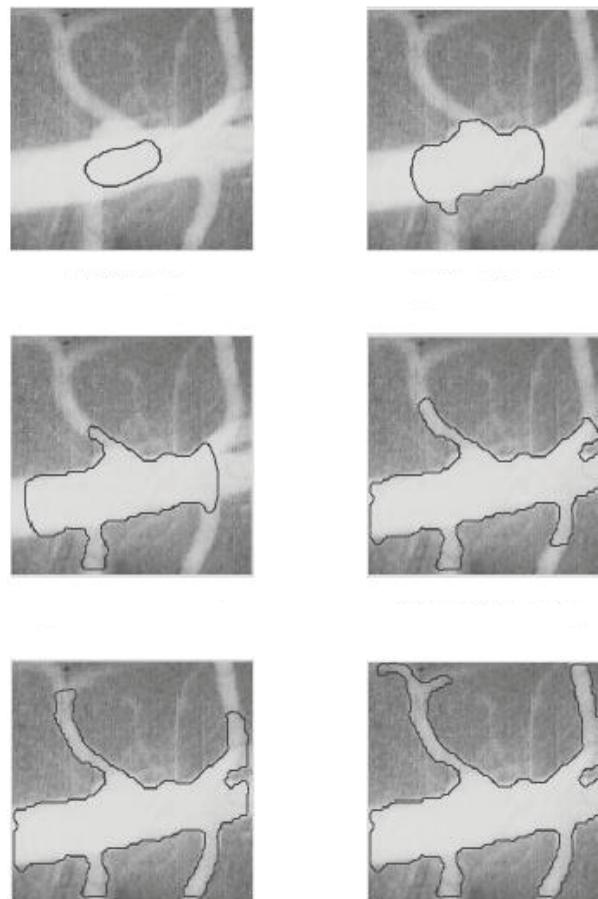
Generalized Cylinder Models

A generalized cylinder (GC) is a solid whose central axis is a 3D curve. Each point of that axis has a limited and closed section that is perpendicular to it. A CG is therefore defined in space by a spatial curve or axis and a function that defines the section in that axis. The section is usually an ellipse. Tecnically, GCs should be included in the parametric methods section, but the work that has been done in this field is so extense that it deserves its own category.

The construction of the coronary tree model requires one single view to build the 2D tree and estimate the sections. However, there is no information on the depth or the area of the sections, so a second projection will be required.

ARTERIAL TRACKING

Contrary to the approaches based on pattern recognition, where local operators are applied to the entire image, techniques based on arterial follow-up are based on the application of local operators in an area that presumibly belongs to a vessel and that cover its length. From a given point of departure the operators detect the central axis and, by analyzing the pixels that are orthogonal to the tracking direction, the vessel's edges. There are various methods to determine the central axis and the edges: some methods carry out a sequential tracking and incorporate connectivity information after a simple edge detection operation, other methods use this information to sequentially track the contours. There are also approaches based on the intensity of the crests, on fuzzy sets, or on the representation of

A

Figure 4. Tracking applied to an angiography
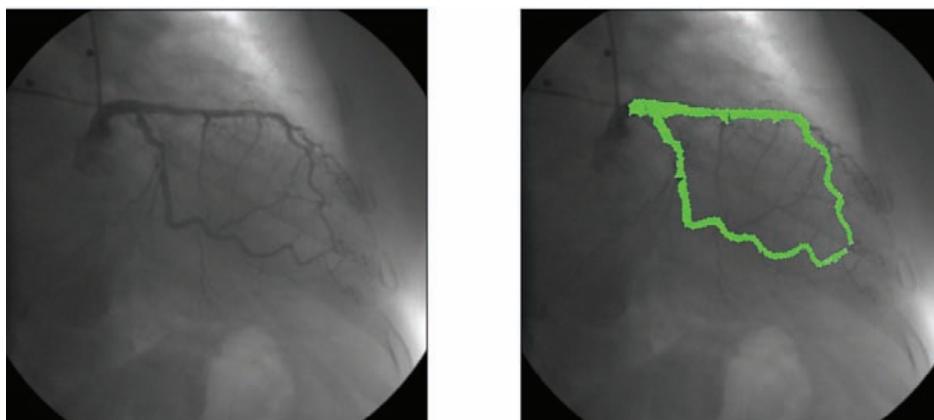
graphs, where the purpose lies in finding the optimal road in the graph that represents the image.

Lu and Eiho (Lu, Eiho, 1993) have described a follow-up algorithm for the vascular edges in **angiographies** that considers the inclusion of branches and consists of three steps:

1. Edge detection
2. Branch search
3. Tracking of sequential contours

The user must provide the point of departure, the direction, and the search range. The edge points are evaluated with a differential smoothening operator in a line that is perpendicular to the direction of the vessel. This operator also serves to detect the branches.

TECHNIQUES BASED ON ARTIFICIAL INTELLIGENCE

Approaches based on Artificial Intelligence use high-level knowledge to guide the **segmentation** and delineation of vascular structures and sometimes use different types of knowledge from various sources.

One possibility (Smets, Verbeeck, Suetens, & Oosterlinck, 1988) is to use rules that codify knowledge on the morphology of blood vessels; these rules are then used to formulate a hierarchy with which to create the model. This type of system does not offer any good results in arterial bifurcations or in arteries with occlusions.

Another approach (Stansfield, 1986) consists in formulating a rules-based **Expert System** to identify the arteries. During the first phase, the image is processed without making use of domain knowledge to extract segments of the vessels. It is only in the second phase that domain knowledge on cardiac anatomy and physiology is applied.

The latter approach is more robust than the former; but it presents the inconvenience of not combining all the segments into one vascular structure.

FUTURE TRENDS

It cannot be said that one technique has a more promising future than another, but the current tendency is to move away from the abovementioned classical **segmentation** algorithms towards 3D and even 4D reconstructions of the coronary tree.

Other lines of research focus on obtaining angiograph images by means of new acquisition technologies such as Magnetic Resonance, **Computarized** High Speed **Tomography**, or two-armed angiograph devices that achieve two simultaneous projections in

combination with the use of ultrasound intravascular devices. This type of acquisition simplifies the creation of tridimensional structures, either directly from the acquisition or after a simple processing of the bidi-mensional images.

REFERENCES

Brown, B. G., Bolson E., Frimer, M., & Dodge, H. T. (1977). Quantitative coronary arteriography. Circula-tion, 55:329-337.

Brown, B. G., Bolson E., Frimer, M., & Dodge, H. T. (1982). Arteriographic assessment of coronary athero-sclerosis. Arteriosclerosis, 2:2-15.

Eiho, S., & Qian, Y. (1997). Detection of coronary artery tree using morphological operator. In Computers in Cardiology 1997, pages 525-528.

Gonzalez, R. C., & Woods, R. E. (1996). Digital Image Proccessing. Addison-Wesley Publishing Company, Inc. Reading, Massachusets, USA.

Greenes, R. A., & Brinkley, K. F. (2001). Imaging Sys-tems. De Medical informatics: computer applications in health care and biomedicine. Pp. 485 – 538. Second Edition. 2001. Ed. Springer-Verlag. New York. USA.

Guo, D., & Richardson, P. (1998) . Automatic vessel extraction from angiogram images. In Computers in Cardiology 1998, 441 - 444.

Jain, R.C., Kasturi, R., & Schunck,B. G. (1995). Ma-chine Vision.McGraw-Hill.

Kirbas, C. & Quek, F. (2004). A review of vessel extraction techniques and algorithms. ACM Comput. Surv., 36(2),81-121.

Klein, A. K., Lee, F., & Amini, A. A. (1997). Quan-titative coronary angiography with deformable spline models. IEEE Transactions on Medical Imaging, 16(5):468-482

Lu, S., & Eiho, S. (1993). Automatic detection of the coronary arterial contours with sub-branches from an x-ray angiogram.In Computers in Cardiology 1993. Proceedings., 575-578.

Nyström, I., Sanniti di Baja, G., & Svensson, S. (2001). Representing volumetric vascular structures using curve skeletons. In Edoardo Ardizzone and Vito Di Gesµu, editors, Proceedings of 11th International Conference on Image Analysis and Processing (ICIAP 2001), 495-500, Palermo, Italy, IEEE Computer Society.

Malladi, R., Sethian, J. A., & Vemuri, B. C. (1995). Shape modeling with front propagation: a level set approach. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 17:158-175.

Mao, F., Ruan, S.,Bruno, A., Toumoulin, C., Collorec, R., & Haigron, P. (1992). Extraction of structural features in digital subtraction angiography. Biomedi-cal Engineering Days, 1992.,Proceedings of the 1992 International, 166-169.

McInerney, T., & Terzopoulos, D.(1997). Medical image segmentation using topologically adaptable surfaces. In CVRMedMRCAS '97: Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery, 23-32, London, UK, Springer-Verlag.

O'Brien, J. F., & Ezquerra, N. F. (1994). Automated segmentation of coronary vessels in angiographic im-age sequences utilizing temporal, spatial and structural constraints. (Technical report), Georgia Institute of Technology.

Pappas, T. N, & Lim, J.S. (1984). Estimation of coronary artery boundaries in angiograms. Appl. Digital Image Processing VII, 504:312-321.

Pellot, C., Herment, A., Sigelle, M., Horain, P., Maitre, H., & Peronneau, P. (1994). A 3d reconstruction of vascular structures from two x-ray angiograms using an adapted simulated annealing algorithm. Medical Imaging, IEEE Transactions on, 13:48-60.

Petrocelli, R. R., Manbeck, K. M., & Elion, J. L. (1993). Three dimensional structure recognition in digital an-giograms using gauss-markov methods. In Computers in Cardiology 1993. Proceedings., 101-104.

Poli, R., & Valli, G. (1997). An algorithm for real-time vessel enhancement and detection. Computer Methods and Programs in Biomedicine, 52:1-22.

Prinet, V., Mona, O., & Rocchisani, J. M. (1995). Multi-dimensional vessels extraction using crest lines. In Engineering in Medicine and Biology Society, 1995. IEEE 17th Annual Conference, 1:393-394.

Quek, F. H. K., & Kirbas, C. (2001). Simulated wave propagation and traceback in vascular extraction. In Medical Imaging and Augmented Reality, 2001. Proceedings. International Worksho, 229-234.

Shmueli, K., Brody, W. R., & Macovski, A. (1983). Estimation of blood vessel boundaries in x-ray images. Opt. Eng., 22:110-116.

Smets, C., Verbeeck, G., Suetens, P., & Oosterlinck, A. (1988). A knowledge-based system for the delineation of blood vessels on subtraction angiograms. Pattern Recogn. Lett., 8(2):113-121.

Stansfield, S. A. (1986). Angy: A rule-based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms. PAMI, 8(3):188-199.

KEY TERMS

Angiography: Image of blood vessels obtained by any possible procedure.

Artery: Each of the vessels that take the blood from the heart to the other bodyparts.

Computerized Tomography: Exploration of X-rays that produces detailed images of axial cuts of the body. A CT obtains many images by rotating around the body. A computer combines all these images into a final image that represents the bodycut like a slice.

Expert System: Computer or computer program that can give responses that are similar to those of an expert.

Segmentation: In computer vision, segmentation refers to the process of partitioning a digital image into multiple regions. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (structures) in images, in this case, the coronary tree in digital angiography frames.

Stenosis: A stenosis is an abnormal narrowing in a blood vessel or other tubular organ or structure. A coronary artery that's constricted or narrowed is called stenosed. Buildup of fat, cholesterol and other substances over time may clog the artery. Many heart attacks are caused by a complete blockage of a vessel in the heart, called a coronary artery.

Thresholding: A technique for the processing of digital images that consists in applying a certain property or operation to those pixels whose intensity value exceeds a defined threshold.

A

ANN Application in the Field of Structural Concrete

Juan L. Pérez
University of A Coruña, Spain

Mª Isabel Martínez
University of A Coruña, Spain

Manuel F. Herrador
University of A Coruña, Spain

INTRODUCTION

Artificial Intelligence (AI) mechanisms are more and more frequently applied to all sorts of civil engineering problems. New methods and algorithms which allow civil engineers to use these techniques in a different way on diverse problems are available or being made available. One AI techniques stands out over the rest: Artificial Neural Networks (ANN). Their most remarkable traits are their ability to learn, the possibility of generalization and their tolerance towards mistakes. These characteristics make their use viable and cost-efficient in any field in general, and in Structural Engineering in particular. The most extended construction material nowadays is concrete, mainly because of its high resistance and its adaptability to formwork during its fabrication process. Along this chapter we will find different applications of ANNs to structural concrete.

Artificial Neural Networks

Warren McCulloch and Walter Pitts are credited for the origin of Artificial Networks in the 1940s, since they were the first to design an artificial neuron (McCulloch & Pitts, 1943). They proposed the binary mode (active or inactive) neuron model with a fixed threshold which must be surpassed for it to change state. Some of the concepts they introduced still hold useful today.

Artificial Neural Networks intend to simulate the properties found in biological neural systems through mathematical models by the way of artificial mechanisms. A neuron is considered a formal element, or module, or basic network unit which receives information from other modules or the environment; it then integrates and computes this information to emit a single output which will be identically transmitted to subsequent multiple neurons (Wasserman, 1989).

The output of an artificial neuron is determined by its propagation or excitation, activation and transfer functions.

The propagation function is generally the summation of each input multiplied by the weight of its interconnection (net value):

$$n_i = \sum_{j=0}^{N-1} \left[W_{ij} \cdot p_j \right] \tag{1}$$

The activation function modifies the latter, relating the neural input to the next activation state.

$$a_i(t) = FA\left[a_i(t-1), n_i(t-1) \right] \tag{2}$$

The transfer function is applied to the result of the activation function. It is used to bound the neuron's output and is generally given by the interpretation intended for the output. Some of the most commonly used transfer functions are the sigmoid (to obtain values in the [0,1] interval) and the hyperbolic tangent (to obtain values in the [-1,1] interval).

$$out_i = FT\left(a_i(t) \right) \tag{3}$$

Once each element in the process is defined, the type of network (network topology) to use must be designed. These can be divided in forward-feed networks, where

A

information moves in one direction only (from input to output), and networks with partial or total feedback, where information can flow in any direction.

Finally, learning rules and training type must be defined. Learning rules are divided in supervised and non-supervised (Brown & Harris, 1994) (Lin & Lee, 1996) and within the latter, self-organizing learning and reinforcement learning (Hoskins & Himmelblau, 1992). The type of training will be determined by the type of learning chosen.

An Introduction to Concrete (Material and Structure)

Structural concrete is a construction material created from the mixture of cement, water, aggregates and additions or admixtures with diverse functions. The goal is to create a material with rock-like appearance, with sufficient compressive strength and the ability to adopt adequate structural shapes. Concrete is moldable during its preparation phase, once the components have mixed together go produce a fluid mass which conveniently occupies the cavities in a mould named formwork. After a few hours, concrete hardens thanks to the chemical hydration reaction experimented by cement, generating a paste which envelops the aggregates and gives the ensemble the appearance of an artificial rock somewhat similar to a conglomerate.

Hardened concrete offers good compressive strength, but very low tensile strength. This is why structures created with this material must be reinforced by use of steel rebars, configured by rods which are placed (before pouring the concrete) along the lines where calculation predicts the highest tensile stresses. Cracking, which reduces the durability of the structure, is thus hindered, and sufficient resistance is guaranteed with a very low probability of failure. The entirety formed by concrete and rebar is referred to as Structural Concrete (Shah, 1993).

Two phases thus characterize the evolution of concrete in time. In the first phase, concrete must be fluid enough to ensure ease of placement, and a time to initial set long enough to allow transportation from plant to worksite. Flowability depends basically on the type and quantity of the ingredients in the mixture. Special chemical admixtures (such as plasticizers and superplasticizers) guarantee flowability without grossly increasing the amount of water, whose ratio relative to the amount of cement (or water/cement ratio, w/c) is on reverse proportion to strength attained. The science of rheology deals with the study of the behavior of fresh concrete. A variety of tests can be used to determine flowability of fresh concrete, the most popular amongst them being the Abrams cone (Abrams, 1922) or slump cone test (Domone, 1998).

The second phase (and longest over time) is the hardened phase of concrete, which determines the behavior of the structure it gives shape to, from the point of view of serviceability (by imposing limitations on cracking and compliance) and resistance to failure (by imposing limitations on the minimal loads that can be resisted, as compared to the internal forces produced by external loading), always within the frame of sufficient durability for the service life foreseen.

The study of structural concrete from every point of view has been undertaken following many different optics. The experimental path has been very productive, generating along the past 50 years a database (with a tendency to scatter) which has been used to sanction studies carried along the second and third path that follow. The analytical path also constitutes a fundamental tool to approach concrete behavior, both from the material and structural point of view. Development of theoretical behavior models goes back to the early 20th century, and theoretical equations developed since have been corrected through testing (as mentioned above) before becoming a part of codes and specifications. This method of analysis has been reinforced with the development of numerical methods and computational systems, capable of solving a great number of simultaneous equations. In particular, the Finite Element Method (and other methods in the same family) and optimization techniques have brought a remarkable capacity to approximate behavior of structural concrete, having their results benchmarked in may applications by the aforementioned experimental testing.

Three basic lines of study are thus available. Being complementary between them, they have played a decisive role in the production of national and international codes and rules which guide or legislate the project, execution and maintenance of structural concrete works. Concrete is a complex material, which presents a number of problems for analytical study, and so is an adequate field for the development of analysis techniques based on neural networks (Gonzalez, Martínez and Carro, 2006)

Application of Artificial Neural Networks to problems in the field of structural concrete has unfolded in the past few years in two ways. On one hand, analytical and structural optimization systems faster than traditional (usually iterative) methods have been generated starting with expressions and calculation rules. On the other, the numerous databases created form the large amount of tests published in the scientific community have allowed for the development of very powerful ANN which have thrown light on various complex phenomena. In a few cases, specific designed codes have been improved through the use of these techniques; some examples follow.

Application of Artificial Neural Networks to Optimization Problems

Design of concrete structures is based on the determination of two basic parameters: member thickness (effective depth d, depth of a beam or slab section measured from the compression face to the centroid of reinforcement) and amount of reinforcement (established as the total area A_s of steel in a section, materialized as rebars, or the reinforcement ratio, the ratio between steel area and concrete area in the section). Calculation methods are iterative, since a large number of conditions must be verified in the structure, and the aforementioned parameters are fixed as a function of three basic conditions which are sequentially followed: structural safety, maximum ductility at failure and minimal cost. Design rules, expressed through equations, allow for a first solution which is corrected to meet all calculation scenarios, finally converging when the difference between input and output parameters are negligible.

In some cases it is possible to develop optimization algorithms, whose analytical formulation opens the way to the generation of a database. Hadi (Hadi, 2003) has performed this work for simply supported reinforced concrete beams, and the expressions obtained after the optimization process determine the parameters specified above, while simultaneously assigning the cost associated to the optimal solution (related to the cost of materials and formwork). With these expressions, Hadi develops a database with the following variables: applied flexural moment (M), compressive strength of concrete (f_c), steel strength (f_y), section width (b), section depth (h), and unit costs of concrete (C_c), steel (C_s) and formwork (C_f).

Network parameters used are as follows. The number of training samples is 550; number of input layer neurons is 8; number of hidden layer neurons is 10; number of output layer neurons is 4; type of backpropagation is Levenberg–Marquardt backpropagation; activation function is sigmoidal function; learning rate; 0.01; number of epochs is 3000; sum-square error achieved is 0.08. The network had been tested with 50 samples and yielded the average error of 6.1%.

Hadi studies various factors when choosing network architecture and backpropagation algorithm type. When two layers of hidden neurons are used, precision is not improved while computation time is increased. The number of samples depends on the complexity of the problem and the number of input and output parameters. If a value is fixed for the input costs, there are no noticeable precision improvements between training the network with 200 or 1000 samples. When costs are introduced as input parameters, 100 samples are not enough to achieve convergence in training. Finally, the training algorithm is also checked, studying the range between pure backpropagation (too slow for training), backpropagation with momentum and with adaptive learning, backpropagation with Levenberg–Marquardt updating rule and fast learning backpropagation. The latter is finally retained since it requires less time to get the network to converge while providing very good results (Demuth, H. & Beale, M.,1995)

Application of Artificial Neural Networks to Prediction of Concrete Physical Parameters Measurable Through Testing: Concrete Strength and Consistency

Other neural network applications are supported by large experimental databases, created through years of research, which allow for the prediction of phenomena with complex analytical formulation.

One of these cases is the determination of two basic concrete parameters: its workability when mixed, necessary for ease of placement in concrete, and its compressive strength once hardened, which is basic to the evaluation of the capacity of the structure. The variables that necessarily determine these two parameters are the components of concrete: amounts of cement, water, fine aggregate (sand), coarse aggregate (small gravel and large gravel), and other components such as pozzolanic additions (which bring soundness

and delayed strength increase, especially in the case of fly ash and silica fume) and admixtures (which fluidify the fresh mixture allowing the use of reduced amounts of water). There are still no analytical or numerical models that faithfully predict fresh concrete consistency (related to flowability, and usually evaluated by the slump of a molded concrete cone) or compressive strength (determined by crushing of prismatic specimens in a press).

Öztaş et al. (Öztaş, Pala, Özbay, Kanca, Çağlar & Batí, 2006) have developed a neural network from 187 concrete mixes, for which all parameters are know, using 169 of them for training and 18, randomly selected, for verification. Database variables are sometimes taken as a ratio between them, since there is available knowledge about the dependency of slump and strength on such parameters. The established range for the 7 parameter set is shown in Table 1.

Network architecture, as determined by 7 input neurons and two hidden layers of 5 and 3 neurons respectively.

The back-propagation learning algorithm has been used in feed-forward two hidden-layers. The learning algorithm used in the study is scaled conjugate gradients algorithm (SCGA), activation function is sigmoidal function, and number of epochs is 10,000. The prediction capacity of the network is better in the "Compressive Strength" output (maximum error of 6%) than in the

"Slump" output (errors up to 25%). This is due to the fact that the relation between the chosen variables and strength is much stronger than in the case of slump, which is influenced by other non-contemplated variables (e. g. type and power of concrete mixer, mixing order of components, aggregate moisture) and the method for measurement of consistency, whose adequacy for the particular type of concrete used in the database is questioned by some authors.

Application of Artificial Neural Networks to the Development of Design Formulae and Codes

The last application presented in this paper is the response analysis to shear forces in concrete beams. These forces generate transverse tensile stresses in concrete beams which require placement of rebars perpendicular to the beam axis, known as hoops or ties. Analytical determination of failure load from the variables that intervene in this problem is very complex, and in general most of the formulae used today are based on experimental interpolations with no dimensional consistency. Cladera and Marí (Cladera & Marí, 2004) have studied the problem through laboratory testing, developing a neural network for the strength analysis of beams with no shear reinforcement. They rely on a database compiled by Bentz (Bentz, 2000) and Kuchma (Kuchma, 2002), where the variables are effective depth (d), beam width (b, though introduced as d/b), shear span (a/d, see Figure 1), longitudinal reinforcement ratio ($\rho_l = A_s/bd$) and compressive strength of concrete (f_c). Of course, failure load is provided for each of the 177 tests found in the database. They use 147 tests to train the network and 30 for verification, on a one layer architecture with 10 hidden neurons and a retropropagation learning mechanism. The ranges

Table 1. Input parameter range

Input parameters	Minimum	Maximum
W/B (ratio, %)[a]	18	45
W (kg/m³)[b]	140	165
s/a (ratio, %)[c]	35	52
FA (ratio, %)[d]	0	20
AE (kg/m³)[e]	0.036	0.078
SF (ratio, %)[f]	5	25
SP (kg/m³)[g]	1.89	36.5

(a) [Water]/[binder] ratio, considering binder as the lump sum of cement, fly ash and silica fume
(b) Amount of water
(c) [Amount of sand]/[Total aggregate (sand+small gravel+large gravel)]
(d) Percentage of cement substituted by fly ash
(e) Amount of air-entraining agent
(f) Percentage of cement substituted by silica fume
(g) Amount of superplasticizer

Table 2 Input parameter ranges

Parameter	Minimum	Maximum
d(mm)	101.6	1090
d/b	0.37	7.17
ρ_l (%)	0.50	6.64
f_c(MPa)	14.7	101.8
a/d	2.48	7.86
V_{fail}(kN)	19.52	332.14

Figure 1. Span loading a of a beam. (González, 2002)
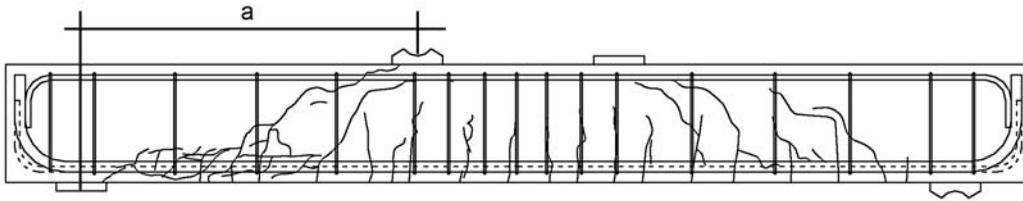
Table 3. Comparison between available codes and proposed equations for shear strength.

Procedure	ACI 11-5	ACI 11-3	MC-90	EC-2	AASHTO	Eq.(7)	Eq.(8)
Average	1.16	1.29	1.15	1.02	1.28	1.15	1.13
Median	1.15	1.25	1.16	0.99	1.25	1.14	1.12
Standard deviation	0.31	0.40	0.19	0.23	0.22	0.18	0.19
CoV (%)	26.89	31.21	16.57	22.03	16.80	15.73	16.42
Minimum	0.42	0.42	0.65	0.57	0.86	0.73	0.78
Maximum	2.14	2.47	1.78	1.78	2.14	1.69	1.85

for the variables are shown on Table 2. Almost 8000 iterations were required to attain best results.

The adjustment provided by training presents an average ratio V_{test}/V_{pred} of 0.99, and 1.02 in validation. The authors have effectively created a laboratory with a neural network, in which they "test" (within parameter range) new beams by changing exclusively one parameter each time. Finally, they come up with two alternative design formulae that improve noticeably any given formula developed up to that moment. Table 3 presents a comparison between those two expressions (named Eq. 7 and Eq. 8) and others found in a series of international codes.

CONCLUSION

- The field of structural concrete shows great potential for the application of neural networks. Successful approaches to optimization, prediction of complex physical parameters and design formulae development have been presented.
- The network topology used in most cases for structural concrete is forward-feed, multilayer with backpropagation, typically with one or two hidden layers. The most commonly used training algorithms are descent gradient with momentum and adaptive learning, and Levenberg-Marquardt.
- The biggest potential of ANNs is their capacity to generate virtual testing laboratories which substitute with precision expensive real laboratory tests within the proper range of values. A methodical "testing" program throws light on the influence of the different variables in complex phenomena at reduced cost.
- The field of structural concrete counts upon extensive databases, generated through the years, that can be analyzed with this technique. An effort should be made to compile and homogenize these databases to extract the maximum possible knowledge, which has great influence on structural safety.

ACKNOWLEDGMENT

A

REFERENCES

Abrams, D.A. (1922). Proportion Concrete Mixtures. Proceedings of the American Concrete Institute, 174-181.

Bentz, EC. (2000). Sectional analysis of reinforced concrete members. PhD thesis, Department of Civil Engineering, University of Toronto.

Brown, M. & Harris, C. (1994). Neurofuzzy adaptive modelling and control. Prentice-Hall.

Cladera, A. & Marí, A.R. (2004). Shear design procedure for reinforced normal and high-strength concrete beams using artificial neural networks. Part I: beams without stirrups. Engineering Structures (26) 917–926

Demuth, H. & Beale, M. (1995). Neural network toolbox for use with MATLAB. MA: The Mathworks, Inc.

Domone, P.(1998). The Slump Flow Test for High-Workability Concrete. Cement and Concrete Research (28-2), 177-182.

González B. (2002). Hormigones con áridos reciclados procedentes de demoliciones: dosificaciones, propiedades mecánicas y comportamiento estructural a cortante. PhD thesis, Department of Construction Technology, University of A Coruña.

González, B. Martínez, I. and Carro, D. (2006). Prediction of the consistency of concrete by means of the use of ANN. Artificial Neural Networks in Real-Life Applications. Ed. Idea Group Inc. 188-200

Hadi, M (2003). Neural networks applications in concrete structures. Computers and Structures (81) 373–381

Hoskins, J.C. & Himmelblau, D.M.(1992). Process control via artificial neural networks and reinforcement learning. Computers and Chemical Engineering, vol. 16(4). 241-251.

Kuchma D. (1999-2002) Shear data bank. University of Illinois, Urbana-Champaign.

Lin, C.T. & Lee, C.S.(1996). Neural Fuzzy Systems: A neuro-fuzzy synergism to intelligent systems. Prentice-Hall.

McCulloch, W. S. & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. (5). 115-133.

Öztaş, A. Pala, M. Özbay E. Kanca E. Çağlar N. & Bhatti M.A. (2006) Predicting the compressive strength and slump of high strength concrete using neural network. Construction and Building Materials. (20). 769–775.

Shah, SP. (1993). Recent trends in the science and technology of concrete, concrete technology, new trends, industrial applications. Proceedings of the international RILEM workshop, London, E & FN Spon. 1–18.

Wasserman, P. (1989) Neural Computing, Ed. Van Nostrand Reinhold, New York.

KEY TERMS

Compression: Stress generated by pressing or squeezing.

Consistency: The relative mobility or ability of freshly mixed concrete or mortar to flow; the usual measurement for concrete is *slump*, equal to the subsidence measured to the nearest 1/4 in. (6 mm) of a molded specimen immediately after removal of the slump cone.

Ductility: That property of a material by virtue of which it may undergo large permanent deformation without rupture.

Formwork: Total system of support for freshly placed concrete including the mold or sheathing that contacts the concrete as well as supporting members, hardware, and necessary bracing; sometimes called *shuttering* in the UK.

Shear Span: Distance between a reaction and the nearest load point.

Structural Safety: Structural response stronger than the internal forces produced by external loading.

Tension: Stress generated by stretching.

ANN Development with EC Tools: An Overview

Daniel Rivero
University of A Coruña, Spain

Juan Rabuñal
University of A Coruña, Spain

INTRODUCTION

Among all of the Artificial Intelligence techniques, Artificial Neural Networks (ANNs) have shown to be a very powerful tool (McCulloch & Pitts, 1943) (Haykin, 1999). This technique is very versatile and therefore has been succesfully applied to many different disciplines (classification, clustering, regression, modellization, etc.) (Rabuñal & Dorado, 2005).

However, one of the greatest problems when using ANNs is the great manual effort that has to be done in their development. A big myth of ANNs is that they are easy to work with and their development is almost automatically done. This development process can be divided into two parts: architecture development and training and validation. As the network architecture is problem-dependant, the design process of this architecture used to be manually performed, meaning that the expert had to test different architectures and train them until finding the one that achieved best results after the training process. The manual nature of the described process determines its slow performance although the training part is completely automated due to the existence of several algorithms that perform this part.

With the creation of Evolutionary Computation (EC) tools, researchers have worked on the application of these techniques to the development of algorithms for automatically creating and training ANNs so the whole process (or, at least, a great part of it) can be automatically performed by computers and therefore few human efforts has to be done in this process.

BACKGROUND

EC is a set of tools based on the imitation of the natural behaviour of the living beings for solving optimization problems. One of the most typical subset of tools inside

EC is called Evolutionary Algorithms (EAs), which are based on natural evolution and its implementation on computers. All of these tools work with the same basis: a population of solutions to that particular problem is randomly created and an evolutionary process is applied to it. From this initial random population, the evolution is done by means of selection and combination of the best individuals (although the worst ones also have a small probability of being chosen) to create new solutions. This process is carried out by selection, crossover, and mutation operators. These operators are typically used in biology in its evolution for adaptation and survival. After several generations, it is hoped that the population contains a good solution to the problem.

The first EA to appear was Genetic Algorithms (GAs), in 1975 (Holland, 1975). With the working explained above, GAs use a binary codification (i.e., each solution is codified into a string of bits). Later, in the early 90s a new technique appeared, called Genetic Programming (GP). This one is based ob the evolution of trees, i.e., each individual is codified as a tree instead of a binary string. This allows its application to a wider set of environments.

Although GAs and GP are the two most used techniques in EAs, more tools can be classified as part of this world, such as Evolutionary Programming or Evolution Strategies, all of them with the same basis: the evolution of a population following the natural evolution rules.

DEVELOPMENT OF ANNS WITH EC TOOLS

The development of ANNs is a topic that has been extensively dealt with very diverse techniques. The world of evolutionary algorithms is not an exception, and proof of that is the great amount of works that have

been published about different techniques in this area (Cantú-Paz & Kamath, 2005). These techniques follow the general strategy of an evolutionary algorithm: an initial population consisting of different genotypes, each one of them codifying different parameters (typically, the weight of the connections and / or the architecture of the network and / or the learning rules), and is randomly created. This population is evaluated in order to determine the fitness of each individual. Afterwards, this population is repeatedly made to evolve by means of different genetic operators (replication, crossover, mutation, etc.) until a determined termination criteria is fulfilled (for example, a sufficiently good individual is obtained, or a predetermined maximum number of generations is achieved).

Essentially, the ANN generation process by means of evolutionary algorithms is divided into three main groups: evolution of the weights, architectures, and learning rules.

Evolution of Weights

The evolution of the weights begins with a network with a predetermined topology. In this case, the problem is to establish, by means of training, the values of the network connection weights. This is generally conceived as a problem of minimization of the network error, taken, for example, as the result of the Mean Square Error of the network between the desired outputs and the ones achieved by the network. Most the training algorithms, such as the backpropagation algorithm (BP) (Rumelhart, Hinton & Williams, 1986), are based on gradient minimization. This has several drawbacks (Whitley, Starkweather & Bogart, 1990), the most important is that quite frequently the algorithm becomes stuck in a local minimum of the error function and is unable of finding the global minimum, especially if the error function is multimodal and / or non-differentiable. One way of overcoming these problems is to carry out the training by means of an Evolutionary Algorithm (Whitley, Starkweather & Bogart, 1990); i.e., formulate the training process as the evolution of the weights in an environment defined by the network architecture and the task to be done (the problem to be solved). In these cases, the weights can be represented in the individuals' genetic material as a string of binary values (Whitley, Starkweather & Bogart, 1990) or a string of real numbers (Greenwood, 1997). Traditional genetic algorithms (Holland, 1975) use a genotypic codification

method with the shape of binary strings. In this way, much work has emerged that codifies the values of the weights by means of a concatenation of the binary values which represent them (Whitley, Starkweather & Bogart, 1990). The big advantage of these approximations is their generality and that they are very simple to apply, i.e., it is very easy and quick to apply the operators of uniform crossover and mutation on a binary string. The disadvantage of using this type of codification is the problem of permutation. This problem was raised upon considering that the order in which the weights are taken in the string causes equivalent networks to possibly correspond with totally different individuals. This leads the crossing operator to become very inefficient. Logically, the weight value codification has also emerged in the form of real number concatenation, each one of them associated with a determined weight (Greenwood 1997). By means of genetic operators designed to work with this type of codification, and given that the existing ones for bit string cannot be used here, several studies (Montana & Davis, 1989) showed that this type of codification produces better results and with more efficiency and scalability than the BP algorithm.

Evolution of the Architectures

The evolution of the architectures includes the generation of the topological structure; i.e., the topology and connectivity of the neurons, and the transfer function of each neuron of the network. The architecture of a network has a great importance in order to successfully apply the ANNs, as the architecture has a very significant impact on the process capacity of the network. In this way, on one hand, a network with few connections and a lineal transfer function may not be able to resolve a problem that another network having other characteristics (distinct number of neurons, connections or types of functions) would be able to resolve. On the other hand, a network having a high number of non-lineal connections and nodes could be overfitted and learn the noise which is present in the training as an inherent part of it, without being able to discriminate between them, and in the end, not have a good generalization capacity. Therefore, the design of a network is crucial, and this task is classically carried out by human experts using their own experience, based on "trial and error", experimenting with a different set of architectures. The evolution of architectures has

been possible thanks to the appearance of constructive and destructive algorithms (Sietsma & Dow, 1991). In general terms, a constructive algorithm begins with a minimum network (with a small number of layers, neurons and connections) and successively adds new layers, nodes and connections, if they are necessary, during the training. A destructive algorithm carries out the opposite operation, i.e., it begins with a maximum network and eliminates unnecessary nodes and connections during the training. However, the methods based on Hill Climbing algorithms are quite susceptible into falling to a local minimum (Angeline, Suders & Pollack, 1994).

In order to develop ANN architectures by means of an evolutionary algorithm, it is necessary to decide how to codify a network inside the genotype so it can be used by the genetic operators. For this, different types of network codifications have emerged.

In the first codification method, direct codification, there is a one-to-one correspondence between the genes and the phenotypic representation (Miller, Todd & Hedge, 1989). The most typical codification method consists of a matrix $C = (c_{ij})$ of NxN size which represents an architecture of N nodes, where c_{ij} indicates the presence or absence of a connection between the i and j nodes. It is possible to use $c_{ij}=1$ to indicate a connection and $c_{ij}=0$ to indicate an absence of connection. In fact, c_{ij} could take real values instead of Booleans to represent the value of the connection weight between neuron "i" and "j", and in this way, architecture and connections can be developed simultaneously (Alba, Aldana & Troya, 1993). The restrictions which are required in the architectures can easily be incorporated into this representational scheme. For example, a feedforward network would have non-zero coefficients only in the upper right hand triangle of the matrix. These types of codification are generally very simple and easy to implement. However, they have a lot of disadvantages, such as scalability, the impossibility of codifying repeated structures, or permutation (i.e., different networks which are functionally equivalent can correspond with different genotypes) (Yao & Liu, 1998).

As a counterproposal to this type of direct codification method, there are also the indirect codification types in existence. With the objective of reducing the length of the genotypes, only some of the characteristics of the architecture are codified into the chromosome. Within this type of codification, there are various types of representation.

First, the parametric representations have to be mentioned. The network can be represented by a set of parameters such as the number of hidden layers, the number of connections between two layers, etc. There are several ways of codifying these parameters inside the chromosome (Harp, Samad & Guha, 1989). Although the parametric representations can reduce the length of the chromosome, the evolutionary algorithm makes a search in a limited space within the possible searchable space that represents all the possible architectures. Another type of non-direct codification is based on a representational system with the shape of grammatical rules (Yao & Shi, 1995). In this system, the network is represented by a set of rules, with shape of production rules, which will build a matrix that represents the network.

Other types of codification, more inspired in the world of biology, are the ones known as "growing methods". With them, the genotype does not codify the network any longer, but instead it contains a set of instructions. The decodification of the genotype consists of the execution of these instructions, which will provoke the construction of the phenotype (Husbands, Harvey, Cliff & Miller, 1994). These instructions usually include neural migrations, neuronal duplication or transformation, and neuronal differentiation.

Finally, and within the indirect codification methods, there are other methods which are very different from the ones already described. Andersen describes a technique in which each individual of a population represents a hidden node instead of the architecture (Andersen & Tsoi, 1993). Each hidden layer is constructed automatically by means of an evolutionary process which uses a genetic algorithm. This method has the limitation that only feed-forward networks can be constructed and there is also a tendency for various nodes with a similar functionality to emerge, which inserts some redundancy inside the network that must be eliminated.

One important characteristic is that, in general, these methods only develop architectures, which is the most common, or else architectures and weights together. The transfer function of each architecture node is assumed to have been previously determined by a human expert, and that it is the same for all of the network nodes (at least, for all of the nodes of the same layer), although the transfer function has been shown to have a great importance on the behaviour of the network (Lovell & Tsoi, 1992). Few methods have

been developed which cause the transfer function to evolve, and, therefore, had little repercussion in the world of ANNs with EC.

Evolution of the Learning Rule

Another interesting approximation to the development of ANNs by means of EC is the evolution of the learning rule. This idea emerges because a training algorithm works differently when it is applied to networks with different architectures. In fact, and given that a priori, the expert usually has very few knowledge about a network, it is preferable to develop an automatic system to adapt the learning rule to the architecture and the problem to be resolved.

There are several approximations to the evolution of the learning rule (Crosher, 1993) (Turney, Whitley & Anderson, 1996), although most of them are based only on how the learning can modify or guide the evolution, and in the relation between the architecture and the connection weights. Actually, there are few works that focus on the evolution of the learning rule in itself (Bengio & Bengio, Cloutier & Gecsei, 1992) (Ribert, Stocker, Lecourtier & Ennaji, 1994).

One of the most common approaches is based on setting the parameters of the BP algorithm: learning rate and momentum. Some authors propose methods in which an evolutionary process is used to find these parameters while leaving the architecture constant (Kim, Jung, Kim & Park, 1996). Other authors, on the other hand, propose codifying these BP algorithm parameters together with the network architecture inside of the individuals of the population (Harp, Samad & Guha, 1989).

FUTURE TRENDS

The evolution of ANNs has been a research topic since some decades ago. The creation of new EC and, in general, new AI techniques and the evolution and improvement of the existing ones allow the development of new methods of automatically developing of ANNs. Although there are methods that (more or less) automatically develop ANNs, they are usually not very efficient, since evolution of architectures, weights and learning rules at once leads to having a very big search space, so this feature definitely has to be improved.

CONCLUSION

The world of EC has provided a set of tools that can be applied to optimization problems. In this case, the problem is to find an optimal architecture and/or weight value set and/or learning rule. Therefore, the development of ANNs was converted into an optimization problem. As the described techniques show, the use of EC techniques has made possible the development of ANNs without human intervention, or, at least, minimising the participation of the expert in this task.

As has been explained, these techniques have some problems. One of them is the already explained permutation problem. Another problem is the loss of efficiency: the more complicated the structure to evolve is (weigths, learning rule, architecture), less efficient the system will be, because the search space becomes much bigger. If the system has to evolve several things at a time (for example, architecture and weights so the ANN development is completely automated), this loss of efficiency increases. However, these systems still work faster than the whole manual process of designing and training several times an ANN.

REFERENCES

Alba E., Aldana J.F. & Troya J.M. (1993) Fully automatic ANN design: A genetic approach. *Proc. Int. Workshop Artificial Neural Networks (IWANN'93), Lecture Notes in Computer Science*. (686) 399-404.

Andersen H.C. & Tsoi A.C. (1993) A constructive algorithm for the training of a multilayer perceptron based on the genetic algorithm. *Complex systems* 7 (4) 249-268.

Angeline P.J., Suders G.M. & Pollack J.B. (1994) An evolutionary algorithm that constructs recurrent neural networks. *IEEE Trans. Neural Networks*. (5) 54-65.

Bengio S., Bengio Y., Cloutier J. & Gecsei J. (1992) On the optimization of a synaptic learning rule. *Preprints of the Conference on Optimality in Artificial and Biological Neural Networks*.

Cantú-Paz E. & Kamath C. (2005) An Empirical Comparison of Combinatios of Evolutionary Algorithms and Neural Networks for Classification Problems. *IEEE Transactions on systems, Man and Cybernetics – Part B: Cybernetics*. 915-927.

Crosher D. (1993) The artificial evolution of a generalized class of adaptive processes. *Preprints of AI'93 Workshop on Evolutionary Computation*. 18-36.

Greenwood G.W. (1997) Training partially recurrent neural networks using evolutionary strategies. *IEEE Trans. Speech Audio Processing*. (5) 192-194.

Harp S.A., Samad T. & Guha A. (1989) Toward the genetic synthesis of neural networks. *Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications*. 360-369.

Haykin, S. (1999). *Neural Networks (2nd ed.)*. Englewood Cliffs, NJ: Prentice Hall.

Holland, J.J. (1975) *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Husbands P., Harvey I., Cliff D. & Miller G. (1994) The use of genetic algorithms for the development of sensorimotor control systems. *From Perception to Action*. (P. Gaussier and JD Nicoud, eds.). Los alamitos CA: IEEE Press.

Kim H., Jung S., Kim T. & Park K. (1996) Fast learning method for backpropagation neural network by evolutionary adaptation of learning rates. *Neurocomputing*, 11(1) 101-106.

Lovell D.R. & Tsoi A.C. (2002) *The Performance of the Neocognitron with various S-Cell and C-Cell Transfer Functions*, Intell. Machines Lab., Dep. Elect. Eng., Univ. Queensland, Tech. Rep.

McCulloch W.S., & Pitts, W. (1943) A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*. (5) 115-133.

Miller G.F., Todd P.M. & Hedge S.U. (1989) Designing neural networks using genetic algorithms. *Proceedings of the Third International Conference on Genetic algorithms*. San Mateo, CA: Morgan Kaufmann. 379-384.

Montana D. & David L. (1989) Training feed-forward neural networks using genetic algorithms. *Proc. 11th Int. Joint Conf. Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann. 762-767.

Rabuñal, J.R. & Dorado J. (2005) *Artificial Neural Networks in Real-Life Applications*. Idea Group Inc.

Ribert A., Stocker E., Lecourtier Y. & Ennaji A. (1994) Optimizing a Neural Network Architecture with an Adaptive Parameter Genetic Algorithm. *Lecture Notes in Computer Science*. Springer-Verlag. (1240) 527-535.

Rumelhart D.E., Hinton G.E. & Williams R.J. (1986) Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. D. E. Rumelhart & J.L. McClelland, Eds. Cambridge, MA: MIT Press. (1) 318-362.

Sietsma J. & Dow R. J. F. (1991) Creating Artificial Neural Networks that generalize. *Neural Networks*. (4) 1: 67-79.

Turney P., Whitley D. & Anderson R. (1996) Special issue on the baldwinian effect. *Evolutionary Computation*. 4(3) 213-329.

Whitley D., Starkweather T. & Bogart C. (1990) Genetic algorithms and neural networks: Optimizing connections and connectivity. *Parallel Comput.*, Vol. 14, No 3. 347-361.

Yao X. & Shi Y. (1995) A preliminary study on designing artificial neural networks using co-evolution. *Proc. IEEE Singapore Int. Conf. Intelligence Control and Instrumentation*. 149-154.

Yao X. & Liu Y. (1998) Toward designing artificial neural networks by evolution. *Appl. Math. Computation*. vol. 91, no. 1, 83-90.

KEY TERMS

Artificial Neural Networks: Interconnected set of many simple processing units, commonly called neurons, that use a mathematical model, that represents an input/output relation,

Back-Propagation Algorithm: Supervised learning technique used by ANNs, that iteratively modifies the weights of the connections of the network so the error given by the network after the comparison of the outputs with the desired one decreases.

Evolutionary Computation: Set of Artificial Intelligence techniques used in optimization problems, which are inspired in biologic mechanisms such as natural evolution.

Genetic Programming: Machine learning technique that uses an evolutionary algorithm in order to optimise the population of computer programs according to a fitness function which determines the capability of a program for performing a given task.

Genotype: The representation of an individual on an entire collection of genes which the crossover and mutation operators are applied to.

Phenotype: Expression of the properties coded by the individual's genotype.

Population: Pool of individuals exhibiting equal or similar genome structures, which allows the application of genetic operators.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in.

ANN–Based Defects' Diagnosis of Industrial Optical Devices

Matthieu Voiry
University of Paris, France
SAGEM REOSC, France

Véronique Amarger
University of Paris, France

Joel Bernier
SAGEM REOSC, France

Kurosh Madani
University of Paris, France

INTRODUCTION

A major step for high-quality optical devices faults diagnosis concerns scratches and digs defects detection and characterization in products. These kinds of aesthetic flaws, shaped during different manufacturing steps, could provoke harmful effects on optical devices' functional specificities, as well as on their optical performances by generating undesirable scatter light, which could seriously damage the expected optical features. A reliable diagnosis of these defects becomes therefore a crucial task to ensure products' nominal specification. Moreover, such diagnosis is strongly motivated by manufacturing process correction requirements in order to guarantee mass production quality with the aim of maintaining acceptable production yield.

Unfortunately, detecting and measuring such defects is still a challenging problem in production conditions and the few available automatic control solutions remain ineffective. That's why, in most of cases, the diagnosis is performed on the basis of a human expert based visual inspection of the whole production. However, this conventionally used solution suffers from several acute restrictions related to human operator's intrinsic limitations (reduced sensitivity for very small defects, detection exhaustiveness alteration due to attentiveness shrinkage, operator's tiredness and weariness due to repetitive nature of fault detection and fault diagnosis tasks).

To construct an effective automatic diagnosis system, we propose an approach based on four main operations: defect detection, data extraction, dimensionality reduction and neural classification. The first operation is based on Nomarski microscopy issued imaging. These issued images contain several items which have to be detected and then classified in order to discriminate between "false" defects (correctable defects) and "abiding" (permanent) ones. Indeed, because of industrial environment, a number of correctable defects (like dusts or cleaning marks) are usually present beside the potential "abiding" defects. Relevant features extraction is a key issue to ensure accuracy of neural classification system; first because raw data (images) cannot be exploited and, moreover, because dealing with high dimensional data could affect learning performances of neural network. This article presents the automatic diagnosis system, describing the operations of the different phases. An implementation on real industrial optical devices is carried out and an experiment investigates a MLP artificial neural network based items classification.

BACKGROUND

Today, the only solution which exists to detect and classify optical surfaces' defects is a visual one, carried out by a human expert. The first originality of this work is in the sensor used: Normarski microscopy. Three main advantages distinguishing Nomarski microscopy (known also as "Differential Interference Contrast microscopy" (Bouchareine, 1999) (Chatterjee, 2003))

from other microscopy techniques, have motivated our preference for this imaging technique. The first of them is related to the higher sensitivity of this technique comparing to the other classical microscopy techniques (Dark Field, Bright Field) (Flewitt & Wild, 1994). Furthermore, the DIC microscopy is robust regarding lighting non-homogeneity. Finally, this technology provides information relative to depth (3-th dimension) which could be exploited to typify roughness or defect's depth. This last advantage offers precious additional potentiality to characterize scratches and digs flaws in high-tech optical devices. Therefore, Nomarski microscopy seems to be a suitable technique to detect surface imperfections.

On the other hand, since they have shown many attractive features in complex pattern recognition and classification tasks (Zhang, 2000) (Egmont-Petersen, de Ridder, & Handels, 2002), artificial neural network based techniques are used to solve difficult problems. In our particular case, the problem is related to the classification of small defects on a great observation's surface. These promising techniques could however encounter difficulties when dealing with high dimensional data. That's why we are also interested in data dimensionality reducing methods.

DEFECTS' DETECTION AND CLASSIFICATION

The suggested diagnosis process is described in broad outline in the diagram of Figure 1. Every step is presented, first detection and data extraction phases and then classification phase coupled with dimensionality reduction. In a second part, some investigations on real industrial data are carried out and the obtained results are presented.

Detection and Data Extraction

The aim of defect's detection stage is to extract defects images from DIC detector issued digital image. The

proposed method (Voiry, Houbre, Amarger, & Madani, 2005) includes four phases:

- Pre-processing: DIC issued digital image transformation in order to reduce lighting heterogeneity influence and to enhance the aimed defects' visibility,
- Adaptive matching: adaptive process to match defects,
- Filtering and segmentation: noise removal and defects' outlines characterization.
- Defect image extraction: correct defect representation construction.

Finally, the image associated to a given detected gives an isolated (from other items) representation of the defect (e.g. depicts the defect in its immediate environment), like depicted in Figure 2.

But, information contained in such generated images is highly redundant and these images don't have necessarily the same dimension (typically this dimension can turn out to be hundred times as high). That is why this raw data (images) can not be directly processed and has first to be appropriately encoded, using some transformations. Such ones must naturally be invariant with regard to geometric transformations (translation, rotation and scaling) and robust regarding different perturbations (noise, luminance variation and background variation). Fourier-Mellin transformation is used as it provides invariant descriptors, which are considered to have good coding capacity in classification tasks (Choksuriwong, Laurent, & Emile, 2005) (Derrode, 1999) (Ghorbel, 1994). Finally, the processed features have to be normalized, using the centring-reducing transformation. Providing a set of 13 features using such transform, is a first acceptable compromise between industrial environment real-time processing constraints and defect image representation quality (Voiry, Madani, Amarger, & Houbre, 2006).

Figure 1. Block diagram of the proposed defect diagnosis system

A

Figure 2. Images of characteristic items: (a) Scratch; (b) dig; (c) dust; (d) cleaning marks

| (a) | (b) | (c) | (d) |
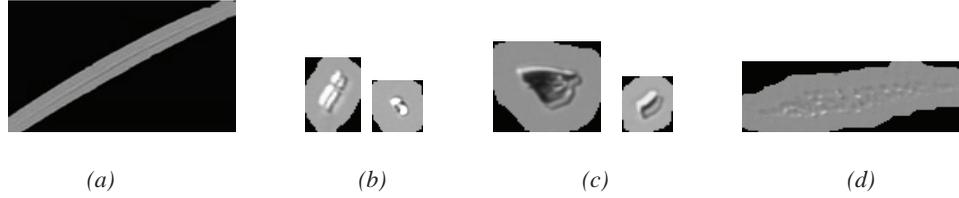
Dimensionality Reduction

To obtain a correct description of defects, we must consider more or less important number of Fourier-Mellin invariants. But dealing with high-dimensional data poses problems, known as "curse of dimensionality" (Verleysen, 2001). First, sample number required to reach a predefined level of precision in approximation tasks increases exponentially with dimension. Thus, intuitively, the sample number needed to properly learn problem becomes quickly much too large to be collected by real systems, when dimension of data increases. Moreover surprising phenomena appear when working in high dimension (Demartines, 1994): for example, variance of distances between vectors remains fixed while its average increases with the space dimension, and Gaussian kernel local properties are also lost. These last points explain that behaviour of a number of artificial neural network algorithms could be affected while dealing with high-dimensional data. Fortunately, most real-world problem data are located in a manifold of dimension p (the data intrinsic dimension) much smaller than its raw dimension. Reducing data dimensionality to this smaller value can therefore decrease the problems related to high dimension.

In order to reduce the problem dimensionality, we use Curvilinear Distance Analysis (CDA). This technique is related to Curvilinear Component Analysis (CCA), whose goal is to reproduce the topology of a n-dimension original space in a new p-dimension space (where p<n) without fixing any configuration of the topology (Demartines & Hérault, 1993). To do so, a criterion characterizing the differences between original and projected space topologies is processed:

$$E_{CCA} = \frac{1}{2} \sum_{i} \sum_{j \neq i} (d_{ij}^{n} - d_{ij}^{p})^2 F(d_{ij}^{p})$$

(1)

Where d_{ij}^{n} (respectively d_{ij}^{p}) is the Euclidean distance between vectors x_i and x_j of considered distribution in original space (resp. in projected space), and F is a decreasing function which favours local topology with respect to the global topology. This energy function is minimized by stochastic gradient descent (Demartines & Hérault, 1995):

$$\forall i \neq j, \Delta x_i^p = \alpha(t) \frac{d_{ij}^n - d_{ij}^p}{d_{ij}^p} u(\lambda(t) - d_{ij}^p)(x_i^p - x_j^p),$$

(2)

Where $\alpha : \Re^+ \to [0;1]$ and $\lambda : \Re^+ \to \Re^+$ are two decreasing functions representing respectively a learning parameter and a neighbourhood factor. CCA provides also a similar method to project, in continuous way, new points in the original space onto the projected space, using the knowledge of already projected vectors.

But, since CCA encounters difficulties with unfolding of very non-linear manifolds, an evolution called CDA has been proposed (Lee, Lendasse, Donckers, & Verleysen, 2000). It involves curvilinear distances (in order to better approximate geodesic distances on the considered manifold) instead of Euclidean ones. Curvilinear distances are processed in two steps way. First is built a graph between vectors by considering k-NN, ε, or other neighbourhood, weighted by Euclidean distance between adjacent nodes. Then the curvilinear distance between two vectors is computed as the minimal distance between these vectors in the graph using Dijkstra's algorithm. Finally the original CCA algorithm is applied using processed curvilinear

distances. This algorithm allows dealing with very non-linear manifolds and is much more robust against the choices of α and λ functions.

It has been successfully used as a preliminary step before maximum likelihood classification in (Lennon, Mercier, Mouchot, & Hubert-Moy, 2001) and we have also showed its positive impact on neural network technique based classification performance (Voiry, Madani, Amarger, & Bernier, 2007). In this last paper, we have first demonstrated that a synthetic problem (nevertheless defined from our real industrial data) whose intrinsic dimensionality is two, is better treated by MLP after 2D dimension reduction than in its raw expression. We have also showed that CDA performs better for this problem than CCA and Self Organizing Map pre-processing.

Implementation on Industrial Optical Devices

In order to validate the above-presented concepts and to provide an industrial prototype, an automatic control system has been realized. It involves an Olympus B52 microscope combined with a Corvus stage, which allows scanning an entire optical component (presented in Figure 3). 50x magnification is used, that leads to microscopic 1.77 mm x 1.33 mm fields and 1.28 μm x 1.28 μm sized pixels. The proposed image processing method is applied on-line. A post-processing software enables to collect pieces of a defect that are detected in different microscopic fields (for example pieces of a long scratch) to form only one defect, and to compute an overall cartography of checked device (Figure 3).

These facilities were used to acquire a great number of Nomarski images, from which were extracted defects images using aforementioned technique. Two

experiments called A and B were carried out, using two different optical devices. Table 1 shows the different parameters corresponding to these experiments. It's important to note that, in order to avoid false classes learning, items images depicting microscopic field boundaries or two (or more) different defects were discarded from used database. Furthermore, studied optical devices were not specially cleaned, what accounts for the presence of some dusts and cleaning marks. Items of these two databases were labelled by an expert with two different labels: "dust" (class1) and "other defects" (class -1). Table 1 shows also items repartition between the two defined classes.

Using this experimental set-up, classification experiment was performed. It involved a multilayer perceptron with n input neurons, 35 neurons in one hidden layer, and 2 output neurons (n-35-2) MLP. First this artificial neural network was trained for discrimination task between classes 1 and -1, using database B. This training phase used BFGS (Broyden, Fletcher, Goldfarb, and Shanno) with Bayesian regularization algorithm, and was achieved 5 times. Subsequently, the generalization ability of obtained neural network was processed using database A. Since database A and B issued from different optical devices, such generalization results are significant. Following this procedure, 14 different experiments were conducted with the aim of studying the global classification performance and the impact of CDA dimensionality reduction on this performance. First experiment used original Fourrier-Mellin issued features (13-dimensional), the others used the same features after CDA n-dimensional space reduction (with n varying between 2 and 13). Figure 4 depicts global classification performances (calculated by averaging percentage of well-classified items for the 5 trainings) for the 14 different experiments, as well as

Figure 3. Automatic control system and cartography of a 100mm x 65mm optical device

Table 1. Description of the two databases used for validation experiments

Database	Optical Device	Number of microscopic fields	Corresponding area	Total items number	Class 1 items number	Class -1 items number
A	1	1178	28 cm²	3865	275	3590
B	2	605	14 cm²	1910	184	1726

Figure 4. Classification performances for different CDA issued data dimensionality. Classification performances using raw data (13-dimensional) are also depicted as dotted lines.
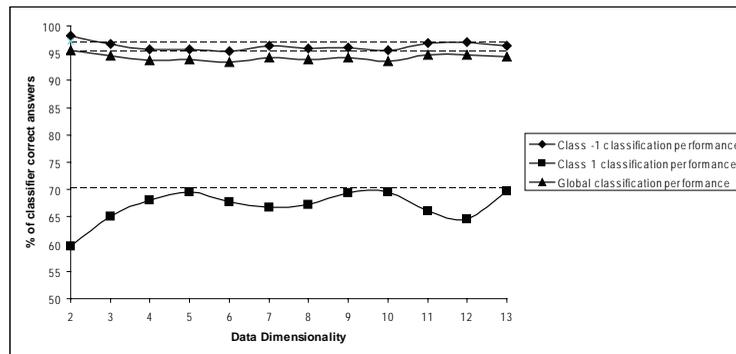
class 1 classification and class -1 classification performances. It shows first that equivalent performances can be obtained using only 5-dimensional data instead of unprocessed defects representations (13-dimensional). As a consequence neural architecture complexity and therefore processing time can be saved using CDA dimensionality reduction, while keeping performance level. Moreover, obtained scores are satisfactory: about 70% of "dust" defects are well-recognized (this can be enough for aimed application) as well as about 97% of other defects (the few 3% errors can however pose problems because every "permanent" defect has to be reported). Furthermore, we think that this significant performances difference between class 1 and class -1 recognition is due to the fact that class 1 is underrepresented in learning database.

FUTURE TRENDS

Next phase of this work will deal with classification tasks involving more classes. We want also use much more Fourier-Mellin invariants, because we think that it would improve classification performance by supplying additional information. In this case, CDA based dimensionality reduction technique would be a foremost step to keep reasonable classification system's complexity and processing time.

CONCLUSION

A reliable diagnosis of aesthetic flaws in high-quality optical devices is a crucial task to ensure products' nominal specification and to enhance the production quality by studying the impact of the process on such defects. To ensure a reliable diagnosis, an automatic system is needed to detect defects and secondly dis-

criminate the "false" defects (correctable defects) from "abiding" (permanent) ones. In this paper is described a complete framework, which allows detecting all defects present in a raw Nomarski image and extracting pertinent features for classification of these defects. Obtained proper performances for "dust" versus "other" defects classification task with MLP neural network has demonstrated the pertinence of proposed approach. In addition, data dimensionality reduction permits to use low complexity classifier (while keeping performance level) and therefore to save processing time.

REFERENCES

Bouchareine, P. (1999). Métrologie des Surfaces. *Techniques de l'Ingénieur, R1390.*

Chatterjee, S. (2003). Design Considerations and Fabrication Techniques of Nomarski Reflection Microscope. *Optical Engineering, 42,* 2202-2212.

Choksuriwong, A., Laurent, H., & Emile, B. (2005). Comparison of invariant descriptors for object recognition. *IEEE International Conference on Image Processing (ICIP)*, 377-380.

Demartines, P. (1994). *Analyse de Données par Réseaux de Neurones Auto-Organisés.* PhD Thesis - Institut National Polytechnique de Grenoble.

Demartines, P. & Hérault, J. (1993). Vector Quantization and Projection Neural Network. *Lecture Notes in Computer Science, 686,* 328-333.

Demartines, P. & Hérault, J. (1995). CCA: "Curvilinear Component Analysis". *Proceedings of 15th workshop GRETSI.*

Derrode, S. (1999). *Représentation de Formes Planes à Niveaux de Gris par Différentes Approximations de Fourier-Mellin Analytique en vue d'Indexation de Bases d'Images.* PhD Thesis - Université de Rennes I.

Egmont-Petersen, M., de Ridder, D., & Handels, H. (2002). Image Processing with Neural Networks - A Review. *Pattern Recognition, 35,* 2279-2301.

Flewitt, P. E. J. & Wild, R. K. (1994). Light Microscopy. In *Physical Methods for materials characterisation*.

Ghorbel, F. (1994). A Complete Invariant Description for Gray Level Images by the Harmonic Analysis Approach. *Pattern Recognition, 15,* 1043-1051.

Lee, J. A., Lendasse, A., Donckers, N., & Verleysen, M. (2000). A Robust Nonlinear Projection Method. In *European Symposium on Artificial Neural Networks - ESANN'2000*.

Lennon, M., Mercier, G., Mouchot, M. C., & Hubert-Moy, L. (2001). Curvilinear Component Analysis for Nonlinear Dimensionality Reduction of Hyperspectral Images. *Proceedings of SPIE, 4541,* 157-168.

Verleysen, M. (2001). Learning high-dimensional data. In *LFTNC'2001 - NATO Advanced Research Workshop on Limitations and Future Trends in Neural Computing*.

Voiry, M., Houbre, F., Amarger, V., & Madani, K. (2005). Toward Surface Imperfections Diagnosis Using Optical Microscopy Imaging in Industrial Environment. *IAR & ACD Workshop 2005 Proceedings*, 139-144.

Voiry, M., Madani, K., Amarger, V., & Bernier, J. (2007). Impact of Data Dimensionality Reduction on Neural Based Classification: Application to Industrial Defects Classification. *Proceedings of the 3rd International Workshop on Artificial Neural Networks and Intelligent Information Processing - ANNIIP 2007*, 56-65.

Voiry, M., Madani, K., Amarger, V., & Houbre, F. (2006). Toward Automatic Defects Clustering in Industrial Production Process Combining Optical Detection and Unsupervised Artificial Neural Network Techniques. *Proceedings of the 2nd International Workshop on Artificial Neural Networks and Intelligent Information Processing - ANNIIP 2006*, 25-34.

Zhang, G. P. (2000). Neural Networks for Classification: A Survey. *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 30,* 451-462.

KEY TERMS

Artificial Neural Networks: A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used

in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Backpropagation algorithm: Learning algorithm of ANNs, based on minimising the error obtained from the comparison between the outputs that the network gives after the application of a set of network inputs and the outputs it should give (the desired outputs).

Classification: Affectation of a phenomenon to a predefined class or category by studying its characteristic features. In our work it consists in determining the nature of detected optical devices surface defects (for example "dust" or "other type of defects").

Data Dimensionality Reduction: Data dimensionality reduction is the transformation of high-dimensional data into a meaningful representation of reduced dimensionality. The goal is to find the important relationships between parameters and reproduce those relationships in a lower dimensionality space. Ideally, the obtained representation has a dimensionality that corresponds to the intrinsic dimensionality of the data. Dimensionality reduction is important in many domains, since it facilitates classification, visualization, and compression of high-dimensional data. In our work it's performed using Curvilinear Distance Analysis.

Data Intrinsic Dimension: When data is described by vectors (sets of characteristic values), data intrinsic dimension is the effective number of degrees of freedom of the vectors' set. Generally, this dimension is smaller than the data raw dimension because it may exist linear and/or non-linear relations between the different components of the vectors.

Data Raw Dimension: When data is described by vectors (sets of characteristic values), data raw dimension is simply the number of components of these vectors.

Detection: Identification of a phenomenon among others from a number of characteristic features or "symptoms". In our work, it consists in identifying surface irregularities on optical devices.

MLP (Multi Layer Perceptron): This widely used artificial neural network employs the perceptron as simple processor. The model of the perceptron, proposed by Rosenblatt is as follows:
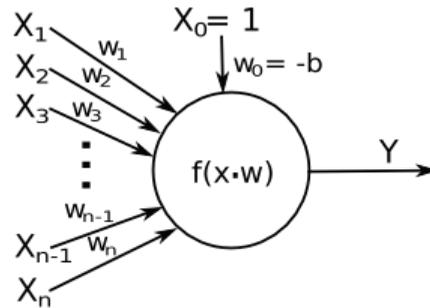
In this diagram, the X represent the inputs and Y the output of the neuron. Each input is multiplied by the weight w, a threshold b is subtracted from the result and finally Y is processed by the application of an activation function f. The weights of the connection are adjusted during a learning phase using backpropagation algorithm.

A

Artificial Intelligence and Education

Eduardo Sánchez
University of Santiago de Compostela, Spain

Manuel Lama
University of Santiago de Compostela, Spain

INTRODUCTION

Governments and institutions are facing the new demands of a rapidly changing society. Among many significant trends, some facts should be considered (Silverstein, 2006): (1) the increment of number and type of students; and (2) the limitations imposed by educational costs and course schedules. About the former, the need of a continuous update of knowledge and competences in an evolving work environment requires life-long learning solutions. An increasing number of young adults are returning to classrooms in order to finish their graduate degrees or attend postgraduate programs to achieve an specialization on a certain domain. About the later, due to the emergence of new types of students, budget constraints and schedule conflicts appear. Workers and immigrants, for instance, are relevant groups for which educational costs and job incompatible schedules could be the key factor to register into a course or to give up a program after investing time and effort on it. In order to solve the needs derived from this social context, new educational approaches should be proposed: (1) to improve and extend the online learning courses, which would reduce student costs and allows to cover the educational needs of a higher number of students, and (2) to automate learning processes, then reducing teacher costs and providing a more personalized educational experience anytime, anywhere.

As a result of this context, in the last decade an increasing interest on applying computer technologies in the field of Education has been observed. On this regard, the paradigms of the Artificial Intelligence (AI) field are attracting an special attention to solve the issues derived from the introduction of computers as supporting resources of different learning strategies. In this paper we review the state-of-art of the application of Artificial Intelligence techniques in the field of Education, focusing on (1) the most popular educational tools based on AI, and (2) the most relevant AI techniques applied on the development of intelligent educational systems.

EXAMPLES OF EDUCATIONAL TOOLS BASED ON AI

The field of Artificial Intelligence can contribute with interesting solutions to the needs of the educational domain (Kennedy, 2002). In what follows, the type of systems that can be built based on AI techniques are outlined.

Intelligent Tutoring Systems

The Intelligent Tutoring Systems are applications that provide personalized/adaptive learning without the intervention of human teachers (VanLehn, 2006). They are constituted by three main components: (1) knowledge of the educational contents, (2) knowledge of the student, and (3) knowledge of the learning procedures and methodologies. These systems promise to radically transform our vision of online learning. As opposed to the hypertext-based e-learning applications, which provide the students with a certain number of opportunities to search for the correct answer before showing it, the intelligent tutoring systems perform like coaches not only after the introduction of the response, but also offering suggestions when the students doubt or are blocked during the process of solving the problem. In this way, the assistance guide the learning process rather than merely saying what is correct or what is wrong.

There exist numerous examples of intelligent tutoring systems, some of them developed at universities as research projects while others created with business goals. Among the first ones, the *Andes* systems (VanLehn, Lynch, Schulze, Shapiro, Shelby, Taylor, Treacy,

Weinstein & Wintersgill, 2005), developed under the guidance of Kurt VanLehn of the University of Pittsburg, is a popular example. The system is in charge of guiding the students while they try to solve different sets of problems and exercises. When the student ask for help in the middle of an activity, the system either provides hints in order to step further towards the solution or points out what was wrong in some earlier step. *Andes* was successfully evaluated during 5 years in the Naval Academy of the United States and can be downloaded for free. Another relevant system is *Cognitive Tutor* (Koedinger, Anderson, Hadley & Mark, 1997), is a comprehensive secondary mathematics curricula and computer-based tutoring program developed by John R. Anderson, professor at the Carnegie Mellon University. The Cognitive Tutor is an example of how research prototypes can be evolved into commercial solutions, as it is nowadays used in 1,500 schools in the United States. On the business side, Read-On! is presented as a product that teaches reading comprehension skills for adults. It analyzes and diagnoses the specific deficiencies and problems of each student and then adapts the learning process based on that features (Read On, 2007). It includes an authoring tool that allows course designers to adapt course contents to different student profiles in a fast and flexible way.

Automatic Evaluation Systems

Automatic Evaluation Systems are mainly focused on evaluating the strengths and weaknesses of students in different learning activities through assessment tests (Conejo, Guzmán, Millan, Trella, Perez-de-la-Cruz. & Rios, 2004). In this way, these systems not only perform the automatic correction of the test, but also derive automatically useful information about the competences and skills obtained by the students during the educational process.

Among the automatic evaluation systems, we could highlight *ToL* (Test On Line) (Tartaglia & Tresso, 2002), which have been used by Physics students in the Polytechnic University of Milano. The system is composed of a database of tests, an algorithm for question selection, and a mechanism for the automatic evaluation of tests, which can be additionally configured by the teachers. *CELLA* (Comprehensive English Language Learning Assesment) (Cella, 2007) is another system that evaluates the student competence on using and understanding the English language. The application

shows the progress carried out by the students and determines their proficiency and degree of competence on the use of foreign languages. As for commercial applications, *Intellimetric* is a Web-based system that lets students to submit their work online (Intellimetric, 2007). In a few seconds, the AI-supported grading engine automatically provides the score of the work. The company claims a reliability of 99%, meaning that 99 percent of the time the engine´s scores match those provided by human teachers.

Computer Supported Collaborative Learning

The environments of computer supported collaborative learning are aimed at facilitating the learning process providing the students both the context and tools to interact and work in a collaborative way with their classmates (Soller, Martinez, Jermann & Muehlenbrock, 2005). In intelligent-based systems, the collaboration is usually carried out with the help of software agents in charge of mediating and supporting student interaction to achieve the proposed learning objectives.

The research prototypes are the suitable test-beds to prove new ideas and concepts, to provide the best collaborative strategies. The *DEGREE* system, for instance, allows the characterization of group behaviours as well as the individual behaviours of the people constituting them, on the basis of a set of attributes or tags. The mediator agent utilizes those attributes, which are introduced by students, in order to provide recommendations and suggestions to improve the interaction inside each group (Barros & Verdejo, 2000). In the business domain there exist multiple solutions although they do not offer intelligent mediation to facilitate the collaborative interactions. The *DEBBIE* system (DePauw Electronic Blackboard for Interactive Education) is one of the most popular (Berque, Johnson, Hutcheson, Jovanovic, Moore, Singer & Slattery, 2000). It was originally developed at the beginning of year 2000 at the University of Depauw, and managed later by the DyKnow company, which was specifically created to make profit with *DEBBIE* (Schnitzler, 2004). The technology that currently offers DyKnow allows both teachers and students to instantaneously share information and ideas. The final goal is to support student tasks in the classroom by eliminating the need of performing simple tasks, as for instance backing up the teacher's presentations. The students could therefore

be more focused on understanding as well as analyzing the concepts presented by the teacher.

Game-Based Learning

Learning based on serious games, a term coined to distinguish between learning-oriented games used in education and purely entertaining-oriented games, deal with the utilization of the motivational power and attractiveness of games in the educational domain in order to improve the satisfaction and performance of students when acquiring new knowledge and skills. This type of learning allows to carry out activities in complex educational environments that would be impossible to implement, because of budget, time, infrastructure and security limitations, with traditional resources (Michael & Chen, 2005; Corti, 2006).

NetAid's is an institution that develop games to teach concepts of global citizenship and to sensitize to fight against poverty. One of its first games, released in 2002, called *NetAid World Class*, consists on taking the identity of a real child living in India and to resolve the real problems that confront the poor children in this region (Stokes, 2005). In 2003 the game was used by 40.000 students in different Schools across the United States. In the business and entertainment arena, many games exist that can be resorted to reach educational goals. Among the most popular ones, *Brain Training* of Nintendo (Brain Training, 2007) challenges the user to improve her mental shape by doing memory, reasoning and mathematical exercises. The final goal is to reach an optimal cerebral age after some regular training.

AI TECHNIQUES IN EDUCATION

The intelligent educational systems reviewed above are based on a diversity of artificial intelligence techniques (Brusilovsky & Peylo, 2003). The most frequently used in the field of education are: (1) personalization mechanisms based on student and group models, (2) intelligent agents and agent-based systems, and (3) ontologies and semantic web techniques.

Personalization Mechanisms

The personalization techniques, which are the basis of intelligent tutoring systems, involve the creation and use of student models. Broadly speaking, these models imply the construction of a qualitative representation of student behavior in terms of existing background knowledge about a domain (McCalla, 1992). These representations can be further used in intelligent tutoring systems, intelligent learning environments, and to develop autonomous intelligent agents that may collaborate with human students during the learning process. The introduction of machine learning techniques facilitates to update and extend the first versions of student models in order to adapt to the evolution of each student as well as the possible changes and modifications of contents and learning activities (Sison & Shimura, 1998). The most popular student modeling techniques are (Beck, Stern, & Haugsjaa, 1996): overlay models and bayesian network models. The first method consists on considering the student model as a subset of the knowledge of an expert in the domain on which the learning is taking place. In fact, the degree of learning is measured in terms of the comparison between the knowledge acquired and represented in the student model with the background initially stored in the expert model. The second method deals with the representation of the learning process as a network of knowledge states. Once defined, the model should infer, from the tutor-student interaction, the probability of the student on being in a certain state.

Intelligent Agents and Agent-Based Systems

Software agents are considered software entities, such as software programs or robots, that present, with different degree, three main attributes: autonomy, cooperation and learning (Nwana, 1996). Autonomy refers to the principle that an agent can operate on their own (acting and deciding upon its own representation of the world). Cooperation refers to the ability to interact with other agents via some communication language. Finally, learning is essential to react or interact with the external environment. Teams of intelligent agents build up MultiAgent Systems (MAS). In this type of systems each agent has either incomplete information or limited capabilities for solving the problem at hand. Other important aspect concerns with the lack of centralized global control; therefore, data is distributed all over the system and computation is asynchronous (Sycara, 1998). Many important tasks can be carried out by intelligent agents in the context of learning and

educational systems (Jafari, 2002, Sánchez, Lama, Amorim, Riera, Vila & Barro, 2003): the monitoring of inputs, outputs, and the activity outcomes produced by the students; the verification of deadlines during homework and exercise submission; automatic answering of student questions; and the automatic grading of tests and surveys.

Ontologies and Semantic Web Techniques

Ontologies aim to capture and represent consensual knowledge in a generic way, and that they may be reused and shared across software applications (Gómez-Pérez, Fernández-López & Corcho, 2004). An ontology is composed of concepts or classes and their attributes, the relationships between concepts, the properties of these relationships, and the axioms and rules that explicitly represents the knowledge of a certain domain. In the educational domain, several ontologies have been proposed: (1) to describe the learning contents of technical documents (Kabel, Wielinga, & de How, 1999), (2) to model the elements required for the design, analysis, and evaluation of the interaction between learners in computer supported cooperative learning (Inaba, Tamura, Ohkubo, Ikeda, Mizoguchi & Toyoda, 2001), (3) to specify the knowledge needed to define new collaborative learning scenarios (Barros, Verdejo, Read & Mizoguchi, 2002), (4) to formalize the semantics of learning objects that are based on metadata standards (Brase & Nejdl, 2004), and (5) to describe the semantics of learning design languages (Amorim, Lama, Sánchez, Riera & Vila, 2006).

FUTURE TRENDS

The next generation of adaptive environments will integrate pedagogical agents, enriched with data mining and machine learning techniques, capable of providing cognitive diagnosis of the learners that will help to determine the state of the learning process and then optimize the selection of personalized learning designs. Moreover, improved models of learners, facilitators, tasks and problem-solving processes, combined with the use of Ontologies and reasoning engines, will facilitate the execution of learning activities on either online platforms or traditional classroom settings.

Research in this field is very active and faces ambitious goals. In some decades it could be possible to dream about sci-fi environments in which the students would have brain interfaces to directly interact with an intelligent assistant (Koch, 2006), which would play the role of a tutor with a direct connection with learning areas of the brain.

CONCLUSION

In this paper we have reviewed the state-of-art of the application of Artificial Intelligence techniques in the field of Education. AI approaches seem promising to improve the quality of the learning process and then to satisfy the new requirements of a rapidly changing society. Current AI-based systems such as intelligent tutoring systems, computer supported collaborative learning and educational games have already proved the possibilities of applying AI techniques. Future applications will both facilitate personalized learning styles and help the tasks of teachers and students in traditional classroom settings.

REFERENCES

Silverstein, S. (2006) Colleges see the future in technology, *Los Angeles Times*.

Kennedy, K. (2002) Top 10 Smart technologies for Schools: Artificial Intelligence. Retrieved from: http://www.techlearning.com/db_area/archives/TL/2002/11/topten5.html

VanLehn, K. (2006) The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 16:227-265.

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor L., Treacy D., Weinstein A. & Wintersgill M. (2005) The Andes Physics Tutoring System: Lessons Learned. *International Journal of Artificial Intelligence in Education*, 15:147-204.

Koedinger K., Anderson J.R., Hadley, W.H. & Mark M.A. (1997) Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education,* 8: 30-43.

Read On! (2007). Retrieved from: http://www.stottlerhenke.com/products/index.htm

Conejo, R., Guzmán, E., Millan, E., Trella, M., Perez-de-la-Cruz, J.L. & Rios, A. (2004) SIETTE: A Web-Based Tool for Adaptive Testing. *International Journal of Artificial Intelligence in Education*, 14:29-61.

Tartaglia A. & Tresso E. (2002) An Automatic Evaluation System for Technical Education at the University Level. IEEE Transactions on Education, 45(3):268-275.

CELLA – Comprehensive English Language Learning Assessment (2007) Retrieved from: http://www.ets.org

Intellimetric (2007) Retrieved from: http://www.vantagelearning.com/intellimetric/

Soller, A., Martinez, A., Jermann, P., & Muehlenbrock, M. (2005) From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. *International Journal of Artificial Intelligence in Education,* 15:261-290.

Barros, B. & Verdejo, M.F. (2000) Analysing student interaction processes in order to improve collaboration: the DEGREE approach. *International Journal of Artificial Intelligence in Education*, 11:221-241.

Berque, D., Johnson, D., Hutcheson, A., Jovanovic, L., Moore, K., Singer, C. & Slattery, K. (2000) The design of an interface for student note annotation in a networked electronic classroom. *Journal of Network and Computer Applications*, 23(2):77-91.

Schnitzler, P. (2004) Becker bets on education software startup. *The Indianapolis Business Journal*, 24(44). Retrieved from: *http://www.dyknowvision.com/news/articles/BeckerBetsOnEdSoftware.html*

Michael, D. & Chen, S. (2005) Serious Games: Games That Educate, Train, and Inform. *Course Technology PTR*.

Corti, K. (2006) Gamesbased Learning: a serious business application. *Report on PixelLearning*. Retrieved from: *http://www.pixelearning.com/docs/serious-gamesbusinessapplications.pdf*

Stokes, B. (2005) Videogames have changed: time to consider Serious Games? *The Development Education Journal*, 11(2).

Brain Training (2007) Retrieved from: *http://es.videogames.games.yahoo.com/especiales/brain-training*

Brusilovsky, P. & Peylo, C. (2003) Adaptive and Intelligent Web-based Educational Systems. *International Journal of Artificial Intelligence in Education*, 13:156–169.

McCalla, G. (1992) The central importance of student modeling to intelligent tutoring. En: *New Directions for Intelligent Tutoring Systems*, Springer Verlag.

Sison, R. & Shimura, M. (1998) Student modeling and machine learning. *International Journal of Artificial Intelligence in Education*, 9:128-158.

Beck, J., Stern, M. & Haugsjaa, E. (1996) Applications of AI in Education. *Crossroads*, 3(1):11-15.

Nwana, H.S. (1996) Software Agents: An Overview. *Knowledge Engineering Review*. 11(2): 205-244.

Sycara, K.P. (1998) Multiagent Systems. *AI Magazine*, 19(2): 79-92.

Jafari, A. (2002) Conceptualizing intelligent agents for teaching and learning. *Educause Quaterly*, 25(3):28-34.

Sánchez, E., Lama, M., Amorim, R., Riera, A., Vila, J & Barro, S. (2003) A multi-tiered agent-based architecture for a cooperative learning environment. *Proceedings of IEEE Euromicro Conference on Parallel and Distributed Processing (PDP 2003)*, 2003.

Gómez-Pérez, A., Fernández-López, M. & Corcho, O. (2004) *Ontological Engineering*, Springer Verlag.

Kabel, S., Wielinga, B. & de How, R. (1999) Ontologies for indexing Technical Manuals for Instruction. *Proceedings of the AIED-Workshop on Ontologies for Intelligent Educational Systems*, LeMans, France, 44-53.

Inaba, A., Tamura, T., Ohkubo, R., Ikeda, M., Mizoguchi, R. & Toyoda, J. (2001) Design and Analysis of Learners Interaction based on Collaborative Learning Ontology. *Proceedings of the Second European Conference on Computer-Supported Collaborative Learning (Euro-CSCL'2001)*, 308-315.

Barros, B., Verdejo, M.F., Read, T. & Mizoguchi, R. (2002) Applications of a Collaborative Learning Ontol-

ogy. *Proceedings of the Second Mexican International Conference on Artificial Intelligence (MICAI 2002)*, 301-310.

Brase, J. & Nejdl, W. (2004) Ontologies and Metadata for eLearning. *Handbook on Ontologies*, Springer-Verlag.

Amorim, R., Lama, M., Sánchez, E., Riera, A. & Vila, X.A. (2006) A Learning Design Ontology based on the IMS Specification. *Journal of Educational Technology & Society*, 9(1):38-57.

Koch, C. (2006) Christof Koch forecast the future. *New Scientist*. Retrieved from: http://www.newscientist.com/channel/opinion/science-forecasts/dn10626-christof-koch-forecasts-the-future.html

KEY TERMS

Automatic Evaluation Systems: Applications focused on evaluating the strengths and weaknesses of students in different learning activities through assessment tests.

Computer Supported Collaborative Learning (CSCL): A research topic on supporting collaborative learning methodologies with the help of computers and collaborative tools.

Game-Based Learning: A new type of learning that combines educational content and computer games in order to improve the satisfaction and performance of students when acquiring new knowledge and skills.

Intelligent Tutoring Systems: A computer program that provides personalized/adaptive instruction to students without the intervention of human beings.

Ontologies: A set of concepts within a domain that capture and represent consensual knowledge in a generic way, and that they may be reused and shared across software applications.

Software Agents: Software entities, such as software programs or robots, characterized by their autonomy, cooperation and learning capabilities.

Student Models: Representation of student behavior and degree of competence in terms of existing background knowledge about a domain.

Artificial Intelligence and Rubble–Mound Breakwater Stability

Gregorio Iglesias Rodriguez
University of Santiago de Compostela, Spain

Alberte Castro Ponte
University of Santiago de Compostela, Spain

Rodrigo Carballo Sanchez
University of Santiago de Compostela, Spain

Miguel Ángel Losada Rodriguez
University of Granada, Spain

INTRODUCTION

Breakwaters are coastal structures constructed to shelter a harbour basin from waves. There are two main types: rubble-mound breakwaters, consisting of various layers of stones or concrete pieces of different sizes (weights), making up a porous mound; and vertical breakwaters, impermeable and monolithic, habitually composed of concrete caissons. This article deals with rubble-mound breakwaters.

A typical rubble-mound breakwater consists of an armour layer, a filter layer and a core. For the breakwater to be stable, the armour layer units (stones or concrete pieces) must not be removed by wave action. Stability is basically achieved by weight. Certain types of concrete pieces are capable of achieving a high degree of interlocking, which contributes to stability by impeding the removal of a single unit.

The forces that an armour unit must withstand under wave action depend on the hydrodynamics on the breakwater slope, which are extremely complex due to wave breaking and the porous nature of the structure. A detailed description of the flow has not been achieved until now, and it is unclear whether it will be in the future in view of the turbulent phenomena involved. Therefore the instantaneous force exerted on an armour unit is not, at least for the time being, amenable to determination by means of a numerical model of the flow. For this reason, empirical formulations are used in rubble-mound design, calibrated on the basis of laboratory tests of model structures. However, these formulations cannot take into account all the aspects affecting the stability, mainly because the inherent complexity of the problem does not lend itself to a simple treatment. Consequently the empirical formulations are used as a predesign tool, and physical model tests in a wave flume of the particular design in question under the pertinent sea climate conditions are *de rigueur*, except for minor structures. The physical model tests naturally integrate all the complexity of the problem. Their drawback lies in that they are expensive and time consuming.

In this article, Artificial Neural Networks are trained and tested with the results of stability tests carried out on a model breakwater. They are shown to reproduce very closely the behaviour of the physical model in the wave flume. Thus an ANN model, if trained and tested with sufficient data, may be used in lieu of the physical model tests. A virtual laboratory of this kind will save time and money with respect to the conventional procedure.

BACKGROUND

Artificial Neural Networks have been used in civil engineering applications for some time, especially in Hydrology (Ranjithan et al., 1993; Fernando and Jayawardena, 1998; Govindaraju and Rao, 2000; Maier and Dandy, 2000; Dawson and Wilby, 2001; Cigizoglu, 2004); some Ocean Engineering issues have also been tackled (Mase et al., 1995; Tsai et al., 2002; Lee and Jeng, 2002; Medina et al., 2003; Kim and Park, 2005; Yagci et al., 2005). Rubble-mound breakwater stabil-

A

ity is studied in Mase et al.'s (1995) pioneering work, focusing on a particular stability formula. Medina et al. (2003) train and test an Artificial Neural Network with stability data from six laboratories. The inputs are the relative wave height, the Iribarren number and a variable representing the laboratory. Kim and Park (2005) compare different ANN models on an analysis revolving around one empirical stability formula, as did Mase et al.'s (1995). Yagci et al. (2005) apply different kinds of neural networks and fuzzy logic, characterising the waves by their height, period and steepness.

PHYSICAL MODEL AND ANN MODEL

The Artificial Neural Networks were trained and tested on the basis of laboratory tests carried out in a wave flume of the CITEEC Laboratory, University of La Coruña. The flume section is 4 m wide and 0.8 m high, with a length of 33 m (Figure 1). Waves are generated by means of a piston-type paddle, controlled by an Active Absorption System (AWACS) which ensures that the waves reflected by the model are absorbed at the paddle.

The model represents a typical three-layer rubble-mound breakwater in 15 m of water, crowned at +9.00 m, at a 1:30 scale. Its slopes are 1:1.50 and 1:1.25

on the seaward and leeward sides, respectively. The armour layer consists in turn of two layers of stones with a weight W=69 g ±10%; those in the upper layer are painted in blue, red and black following horizontal bands, while those in the lower layer are painted in white, in order to easily identify after a test the damaged areas, *i.e.*, the areas where the upper layer has been removed. The filter layer is made up of a gravel with a median size D_{50} = 15.11 mm and a thickness of 4 cm. Finally, the core consists of a finer gravel, with D_{50} = 6.95 mm, D_{15} = 5.45 mm, and D_{85} = 8.73 mm, and a porosity n = 42%. The density of the stones and gravel is γ_r = 2700 kg/m³.

Waves were measured at six different stations along the longitudinal, or x-axis, of the flume. With the origin of x located at the rest position of the wave paddle, the first wave gauge, S1, was located at x=7.98 m. A group of three sensors, S2, S3 and S4, was used to separate the incident and the reflected waves. The central wave gauge, S3, was placed at x=12.28 m, while the position of the others, S2 and S4, was varied according to the wave generation period of each test (Table 1). Another wave gauge, S5, was located 25 cm in front of the model breakwater toe, at x=13.47 m, and 16 cm to the right (as seen from the wave paddle) of the flume centreline, so as not to interfere with the video recording of the

Figure 1. Experimental set-up
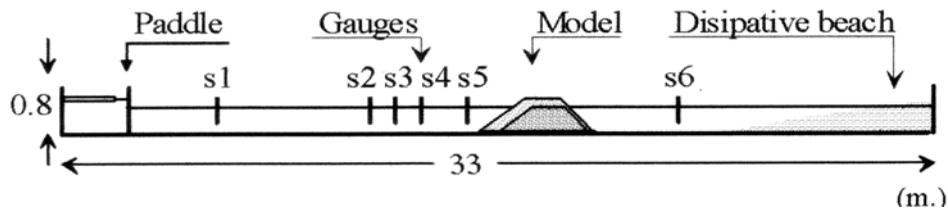
Table 1. Relative water depth (kh), wave period (T), and separation between sensors S2, S3 and S4 in the stability tests

Test key	*kh*	*T* (s)	S2-S3 (cm)	S3-S4 (cm)
T10, T20	0.98	1.65	35	55
T11, T21	1.36	1.3	20	30
T12, T22	1.68	1.13	20	30
T13, T23	1.97	1.03	20	30

tests. Finally, a wave gauge (S6) was placed to the lee of the model breakwater, at x=18.09 m.

Both regular and irregular waves were used in the stability tests. This article is concerned with the eight regular wave tests, carried out with four different wave periods. The water depth in the flume was kept constant throughout the tests (*h*=0.5 m). Each test consisted of a number of wave runs with a constant value of the wave period *T,* related to the wavenumber *k* by

$$T = 2\pi \left[gk \tanh(kh) \right]^{-\frac{1}{2}},$$

where *g* is the gravitational acceleration. The wave periods and relative water depths (*kh*) of the tests are shown in Table 1.

Each wave run consisted of 200 waves. In the first run of each test, the generated waves had a model height *H*=6 cm (corresponding to a wave height in the prototype H_p=1.80 m); in the subsequent runs, the wave height was increased in steps of 1 cm (7 cm, 8 cm, 9 cm, etc.), so that the model breakwater was subject to ever more energetic waves.

Four damage levels (Losada *et al.,* 1986) were used to characterize the stability situation of the model breakwater after each wave run:

(0) No damage. No armour units have been moved from their positions.
(1) Initiation of damage. Five or more armour units have been displaced.
(2) Iribarren damage. The displaced units of the armour's first (outer) layer have left uncovered an area of the second layer large enough for a stone to be removed by waves.
(3) Initiation of destruction. The first unit of the armour's second layer has been removed by wave action.

As the wave height was increased through a test, the damage level also augmented from the initial 'no damage' to 'initiation of damage', 'Iribarren damage', and eventually 'initiation of destruction', at which point the test was terminated and the model rebuilt for the following test. The number of wave runs in a test varied from 10 to 14.

The foregoing damage levels provide a good semi-quantitative assessment of the breakwater stability condition. However, the following nondimensional

damage parameter is more adequate for the Artificial Neural Network model:

$$S = \frac{nD_{50}}{(1-p)b}$$

where D_{50} is the median size of the armour stones, *p* is the porosity of the armour layer, *b* is the width of the model breakwater, and *n* is the number of units displaced after each wave run. In this case, D_{50}= 2.95 cm, *p* = 0.40, and *b* = 50 cm.

The incident wave height was nondimensionalized by means of the zero-damage wave height of the SPM (1984) formulation,

$$H_0 = \left(\frac{W K_D \left(\frac{\gamma_r}{\gamma_w} - 1 \right)^3 \cot\alpha}{\gamma_r} \right)^{\frac{1}{3}}$$

where K_D=4 is the stability coefficient, γ_w=1000 kg/m³ is the water density (freshwater used in the laboratory tests), and α is the breakwater slope. With these values, H_0 = 9.1 cm. The nondimensional incident wave height is given by

$$H^* = \frac{H}{H_0}$$

where *H* stands for the incident wave height.

Most of the previous applications of Artificial Neural Networks in Civil Engineering use multilayer feedforward networks trained with the backpropagation algorithm (Freeman and Skapura, 1991; Johansson et al., 1992), which will also be employed in this study; their main advantage lies in their generalisation capabilities. Thus this kind of network may be used, for instance, to predict the armour damage that a model breakwater will sustain under certain conditions, even if these conditions were not exactly part of the data set with which the network was trained. However, the parameters describing the conditions (e. gr., wave height and period) must be within the parameter ranges of the stability tests with which the ANN was trained.

In this case, the results from the stability tests of the model rubble-mound breakwater described above were used to train and test the Artificial Neural Network. The eight stability tests comprised 96 wave runs. The input to the network was the nondimensional wave height (*H**) and the relative water depth (*kh*) of a wave run, and the output, the resulting nondimensional damage parameter (*S*). Data from 49 wave runs, corresponding to the four stability tests T20, T21, T22, and T23, were used for training the network; while data from 46 wave runs, pertaining to the remaining four tests (T10, T11, T12, and T13) were used for testing it. This distribution of data made sure that each of the four wave generation periods (Table 1) was present in both the training and the testing data sets.

First, an Artificial Neural Network with 10 sigmoid neurons in the hidden layer and a linear output layer was trained and tested 10 times. The ANN was trained by means of the Bayesian Regularisation method (MacKay, 1992), known to be effective in avoiding overfitting. The average MSE values were 0.2880 considering all the data, 0.2224 for the training data set, and 0.3593 for the testing data set. The standard deviations of the MSE values were 5.9651×10^{-10}, 9.0962×10^{-10}, and 7.7356×10^{-10}, for the complete data set, the training and the testing data sets, respectively. Increasing the number of neural units in the hidden layer to 15 did not produce any significant improvement in the average MSE values (0.2879, 0.2222 and 0.3593 for all the data, the training data set and the testing data set, respectively), so the former Artificial Neural Network, with 10 neurons in the hidden layer, was retained.

The following results correspond to a training and testing run of this ANN with a global MSE of 0.2513. The linear regression analysis indicates that the ANN data fit very well to the experimental data over the whole range of the nondimensional damage parameter *S*. In effect, the correlation coefficient is 0.983, and the equation of the best linear fit, $y = 0.938x - 0.00229$, is very close to that of the diagonal line $y = x$ (Figure 2).

The results obtained with the training data set (stability tests T20, T21, T22 and T23) show an excellent agreement between the ANN model and the physical model (Figure 3). In three of the four tests (T20, T22 and T23) the ANN data mimic the measurements on the model breakwater almost to perfection. In test T21, the physical model experiences a brusque increase in the damage level at *H** =1.65, which is slightly softened by the ANN model. The MSE value is 0.1441.

Figure 2. Regression analysis. Complete data set.
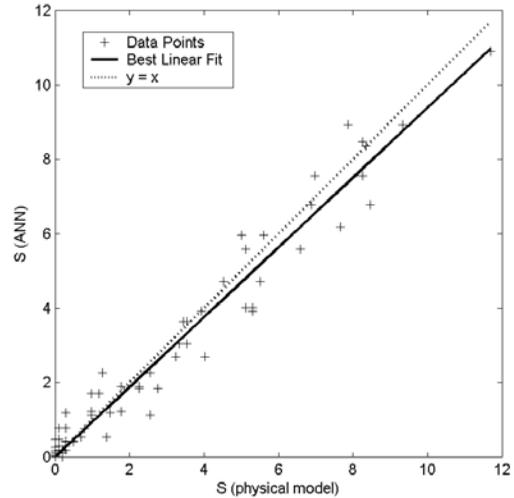
The testing data set comprised also four stability tests (T10, T11, T12 and T13). The inherent difficulty of the problem is apparent in test T11 (Figure 4), in which the nondimensional damage parameter (*S*) does not increase in the wave run at *H** =1.54, but suddenly soars by about 100% in the next wave run, at *H** =1.65. Such differences from one wave run to the next are practically impossible to capture by the ANN model, given that the inputs to the ANN model either vary only slightly, by less than 7% in this case (the nondimensional wave height, *H**) or do not vary at all (the relative water depth, *kh*). It should be remembered that, when computing the damage after a given wave run, the ANN does not have any information about the damage level before that wave run, unlike the physical model. Yet the ANN performs well, yielding an MSE value of 0.3678 with the testing data set.

FUTURE TRENDS

In this study, results from stability tests carried out with regular waves were used. Irregular wave tests should also be analyzed by means of Artificial Intelligence, and it is the authors' intention to do so in the future. Breakwater characteristics are another important aspect of the problem. The ANN cannot extrapolate beyond the ranges of wave and breakwater characteristics on which it was trained. The stability tests used for this

Figure 3. ANN (□) and physical model results (○) for the stability tests T20, T21, T22 and T23 (training data set)
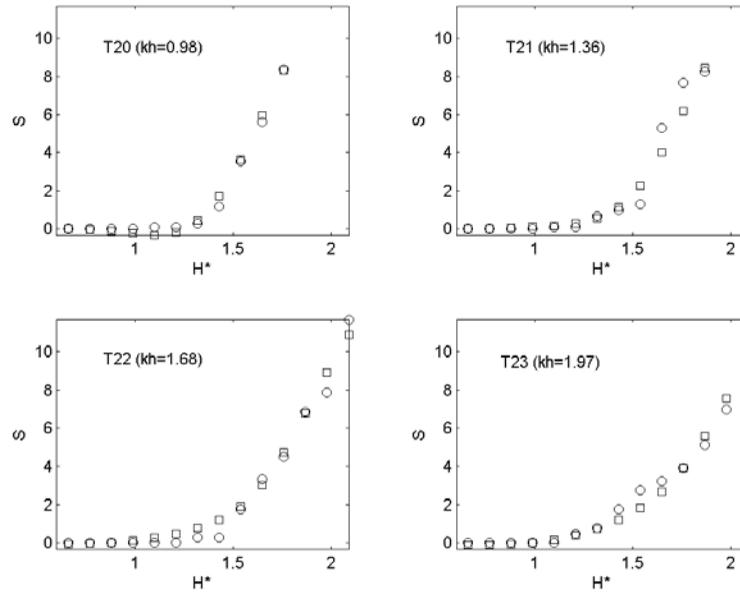
Figure 4. ANN (□) and physical model results (○) for the stability tests T10, T11, T12 and T13 (testing data set)

A
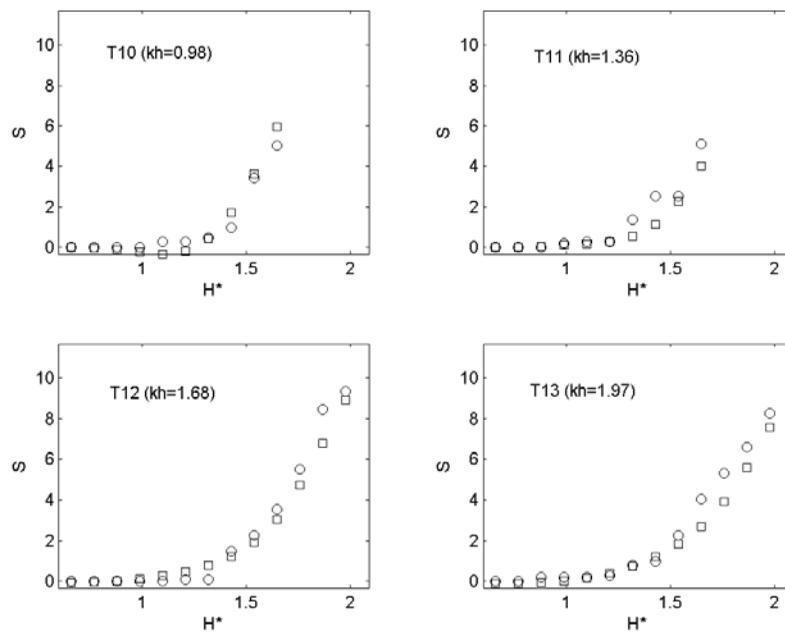
study considered one model breakwater; further tests involving physical models with other geometries and materials should be undertaken. Once the potential of Artificial Neural Networks to model the behaviour of a rubble-mound breakwater subject to wave action has been proven, a virtual laboratory could be constructed with the results from these tests.

CONCLUSION

This article shows that Artificial Neural Networks are capable of modelling the behaviour of a model rubble-mound breakwater in the face of energetic waves. This is a very complex problem for a number of reasons. In the first place, the hydrodynamics of waves breaking on a slope are not well known, so much so that a detailed characterization of the motions of the water particles is not possible for the time being, and may remain so in the future due to the chaotic nature of the processes involved. Second, in the case of a rubble-mound breakwater the problem is further compounded by the porous nature of the structure, which brings about a complex wave-structure interaction in which the flux of energy carried by the incident wave is distributed into the following processes: (i) wave reflection; (ii) wave breaking on the slope; (iii) wave transmission through the porous medium; and (iv) dissipation. The subtle interplay between all these processes means that it is not possible to study one of them without taking the others into account. Third, the porous medium itself is of a stochastic nature: no two rubble-mound breakwaters can be said to be identical. This complexity has precluded up to now the development of a numerical model which can reliably analyse the forces acting on the armour layer units and hence the stability situation of the breakwater. As a consequence, physical model tests are a necessity whenever a major rubble-mound structure is envisaged.

Notwithstanding the difficulty of the problem, the Artificial Neural Network used in this work has been shown to reproduce very closely the physical model results. Thus, an Artificial Neural Network can constitute, once properly trained and validated, a virtual laboratory. Testing a breakwater in this virtual laboratory is much quicker and far less expensive that testing a physical model of the same structure in a laboratory wave flume.

REFERENCES

Cigizoglu, H.K., 2004. Estimation and forecasting of daily suspended sediment data by multilayer perceptrons. Advances in Water Resources 27,185-195.

Dawson, C.W., Wilby, R.L., 2001. Hydrological modeling using artificial neural networks. Progress in Physical Geography 25 (1), 80-108. 20

Fernando, D.A.K., Jayawardena, A.W., 1998. Runoff forecasting using RBF networks with OLS algorithm. Journal of Hydrologic Engineering 3(3), 203-209.

Freeman, J. A., Skapura, D. M., 1991. Neural Networks. Algorithms, Applications, and Programming Techniques. Addison-Wesley.

Govindaraju, R.S., Rao, A.R., 2000. Artificial neural networks in hydrology. Kluwer Academic Publishers, Dordrecth Boston, MA, p. 329.

Haykin, S. (1999). Neural Networks (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Johansson, E. M., Dowla, F. U., Goodman, D. M., 1992. Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method. Int. J. of Neural Systems, 2(4), 291-301.

Kim, D.H., Park, W.S., 2005. Neural network for design and reliability analysis of rubble mound breakwaters. Ocean Engineering 32 (11-12), 1332-1349. 21

Lee, T.L., Jeng, D.S., 2002. Application of artificial neural networks in tideforecasting. Ocean Engineering 29 (9), 1003–1022.

Lippmann, R. P., 1987. An Introduction to Computing with Neural Nets, IEEE, ASSP Magazine.

Losada, M. A., Desiré, J. M., Alejo, L. M., 1986. Stability of blocks as breakwater armor units. J. Struc. Engrg., ASCE, 112(11), 2392-2401.

MacKay, D. J. C., 1992, Bayesian interpolation, Neural Computation, vol. 4, no. 3, pp. 415-447.

Maier, H.R., Dandy, G.C., 2000. Neural network for the prediction and forecasting of water resources variables: a review of modeling issues and applications. Environmental Modeling and Software 15, 101–124.

Mase, H., Sakamoto, M., Sakai, T., 1995. Neural network for stability analysis of rubble mound break-

waters. Journal of Waterway, Port, Coastal and Ocean Engineering, ASCE 121 (6), 294–299.

Medina, J. R., Garrido, J., Gómez-Martín, M.E., Vidal, C., 2003. Armour damage analysis using Neural Networks. Proc. Coastal Structures '03, Portland, Oregon (USA).

Ranjithan, S., Eheart, J. W., Garrett, J.H., 1993. Neural network-based screening for groundwater reclamation under uncertainty. Water Resources Research 29 (3), 563-574.

SPM, 1984. Shore Protection Manual. Dept. of the Army, Coast. Engrg. Res. Ctr., Wtrwy. Experiment Station, Vicksburg, Miss. (USA).

Tsai, C.P., Lin, C., Shen, J.N., 2002. Neural network for wave forecasting among multi-stations. Ocean Engineering 29 (13), 1683–1695.

Yagci, O., Mercan, D.E., Cigizoglu, H.K., Kabdasli, M.S., 2005. Artificial intelligence methods in breakwater damage ratio estimation. Ocean Engineering 32 (17-18), 2088-2106.

KEY TERMS

Armour Damage: Extraction of stones or concrete units from the armour layer by wave action.

Armour Layer: Outer layer of a rubble-mound breakwater, consisting of heavy stones or concrete blocks.

Artificial Neural Networks: Interconnected set of many simple processing units, commonly called neurons, that use a mathematical model representing an input/output relation.

Backpropagation Algorithm: Supervised learning technique used by ANNs that iteratively modifies the weights of the connections of the network so the error given by the network after the comparison of the outputs with the desired one decreases.

Breakwater: Coastal structure built for sheltering an area from waves, usually for loading or unloading vessels.

Reflection: The process by which the energy of the incoming waves is returned seaward.

Significant Wave Height: In wave record analysis, the average height of the highest one-third of a selected number of waves.

Artificial Intelligence for Information Retrieval

Thomas Mandl
University of Hildesheim, Germany

INTRODUCTION

This article describes the most prominent approaches to apply artificial intelligence technologies to **information retrieval** (IR). **Information retrieval** is a key technology for knowledge management. It deals with the search for information and the representation, storage and organization of knowledge. **Information retrieval** is concerned with search processes in which a user needs to identify a subset of information which is relevant for his information need within a large amount of knowledge. The information seeker formulates a query trying to describe his information need. The query is compared to document representations which were extracted during an **indexing** phase. The representations of documents and queries are typically matched by a similarity function such as the Cosine. The most similar documents are presented to the users who can evaluate the relevance with respect to their problem (Belkin, 2000). The problem to properly represent documents and to match imprecise representations has soon led to the application of techniques developed within Artificial Intelligence to information retrieval.

BACKGROUND

In the early days of computer science, **information retrieval** (IR) and artificial intelligence (AI) developed in parallel. In the 1980s, they started to cooperate and the term intelligent **information retrieval** was coined for AI applications in IR. In the 1990s, **information retrieval** has seen a shift from set based Boolean retrieval models to **ranking** systems like the vector space model and **probabilistic** approaches. These approximate reasoning systems opened the door for more intelligent value added components. The large amount of text documents available in professional databases and on the internet has led to a demand for intelligent methods in text retrieval and to considerable research in this area. The need for better preprocessing to extract more knowledge from data has become an important way to improve systems. Off the shelf approaches promise worse results than systems adapted to users, domain and information needs. Today, most techniques developed in AI have been applied to retrieval systems with more or less success. When data from users is available, systems use often machine learning to optimize their results.

Artificial Intelligence Methods in Information Retrieval

Artificial intelligence methods are employed throughout the standard **information retrieval** process and for novel value added services. The first section gives a brief overview of information retrieval. The subsequent sections are organized along the steps in the retrieval process and give examples for applications.

Information Retrieval

Information retrieval deals with the storage and **representation** of knowledge and the retrieval of information relevant for a specific user problem. The information seeker formulates a query trying to describe his information need. The query is compared to document **representations**. The representations of documents and queries are typically matched by a similarity function such as the Cosine or the Dice coefficient. The most similar documents are presented to the users who can evaluate the relevance with respect to their problem.

Indexing usually consists of the several phases. After word segmentation, stopwords are removed. These common words like articles or prepositions contain little meaning by themselves and are ignored in the document **representation**. Second, word forms are transformed into their basic form, the stem. During the **stemming** phase, e.g. houses would be transformed into house. For the document **representation**, different word forms are usually not necessary. The importance of a word for a document can be different. Some words better describe the content of a document than others.

This weight is determined by the frequency of a stem within the text of a document (Savoy, 2003).

In multimedia retrieval, the context is essential for the selection of a form of query and document **representation**. Different media **representation**s may be matched against each other or transformations may become necessary (e.g. to match terms against pictures or spoken language utterances against documents in written text).

As **information retrieval** needs to deal with vague knowledge, exact processing methods are not appropriate. Vague retrieval models like the **probabilistic** model are more suitable. Within these models, terms are provided with weights corresponding to their importance for a document. These weights mirror different levels of relevance.

The result of current **information retrieval** systems are usually sorted lists of documents where the top results are more likely to be relevant according to the system. In some approaches, the user can judge the documents returned to him and tell the systems which ones are relevant for him. The system then resorts the result set. Documents which contain many of the words present in the relevant documents are ranked higher. This relevance feedback process is known to greatly improve the performance. Relevance feedback is also an interesting application for machine learning. Based on a human decisions, the optimization step can be modeled with several approaches, e.g. with rough sets (Singh & Dey 2005). In Web environments, a click is often interpreted as an implicit positive relevance judgment (Joachims & Radlinski, 2007).

Advanced Representation Models

In order to represent documents in natural language, the content of these documents needs to be analyzed. This is a hard task for computer systems. Robust semantic analysis for large text collections or even multimedia objects has yet to be developed. Therefore, text documents are represented by natural language terms mostly without syntactic or semantic context. This is often referred to as the bag-of-words approach. These keywords or terms can only imperfectly represent an object because their context and relations to other terms are lost.

However, great progress has been made and systems for semantic analysis are getting competitive. Advanced syntactic and semantic parsing for robust processing of mass data has been derived from computational linguistics (Hartrumpf, 2006).

For application and domain specific knowledge, another approach is taken to improve the **representation** of documents. The **representation** scheme is enriched by exploiting knowledge about concepts of the domain (Lin & Demner-Fushman, 2006).

Match Between Query and Document

Once the **representation** has been derived, a crucial aspect of an **information retrieval** system is the similarity calculation between query and document **representation**. Most systems use mathematical similarity functions such as the Cosine. The decision for a specific function is based on heuristics or empirical evaluations. Several approaches use machine learning for long term optimization of the matching between term and document. E.g. one approach applies genetic algorithm to adapt a weighting function to a collection (Almeida et al., 2007).

Neural networks have been applied widely in IR. Several network architectures have been applied for retrieval tasks, most often the so-called spreading activation networks are used. Spreading activation networks are simple Hopfield-style networks, however, they do not use the learning rule of Hopfield networks. They typically consist of two layers representing terms and documents. The weights of connections between the layers are bi-directional and initially set according to the results of the traditional **indexing** and weighting algorithms (Belkin, 2000). The neurons corresponding to the terms of the user's query are activated in the term layer and activation spreads along the weights into the document layer and back. Activation represents relevance or interest and reaches potentially relevant terms and documents. The most highly activated documents are presented to the user as result. A closer look at the models reveals that they very much resemble the traditional vector space model of **Information Retrieval** (Mandl, 2000). It is not until after the second step that associative nature of the spreading activation process leads to results different from a vector space model. The spreading activation networks successfully tested with mass data do not take advantage of this associative property. In some systems the process is halted after only one step from the term layer into the document layer, whereas others make one more step

back to the term layer to facilitate learning (Kwok & Grunfeld, 1996).

Queries in **information retrieval** systems are usually short and contain few words. Longer queries have a higher probability to achieve good results. As a consequence, systems try to add good terms to a query entered by a user. Several techniques have been applied. Either these terms are taken from top ranked documents or terms similar to the original ones are used. Another technique is to use terms from documents from the same category. For this task, classification algorithms from machine learning are used (Sebastiani, 2002).

Link analysis applies well known measures from bibliometric analysis to the Web. The number links pointing to a Web page is used as an indicator for its quality (Borodin et al., 2005). PageRank assigns an authority value to each Web page which is primarily a function of its back links. Additionally, it assumes that links from pages with high authority should be weighed higher and should result in a higher authority for the receiving page. To account for the different values each page has to distribute, the algorithm is carried out iteratively until the result converges (Borodin et al., 2005). Machine Learning approaches complement link analysis. Decisions of humans about the quality of Web pages are used to determine design features of these pages which are good indicators of their quality. Machine learning models are applied to determine the quality of pages not judged yet (Mandl, 2006, Marti & Hearst, 2002).

Learning from users has been an important strategy to improve systems. In addition to the content, artificial intelligence methods have been used to improve the user interface.

Value Added Components for User Interfaces

Several Researchers have implemented **information retrieval** systems based on the Kohonen **self organizing map** (SOM), a neural network model for unsupervised classification. They provide an associative user interface where neighborhood of documents expresses a semantic relation. Implementations for large collections can be tested on the internet (Kohonen, 1998). The SOM consists of a usually two-dimensional grid of neurons, each associated with a weight vector. Input documents are classified according to the similarity between the input pattern and the weight vectors, and,

the algorithm adapts the weights of the winning neuron and its neighbor. In that way, neighboring clusters have a high similarity.

The **information retrieval** applications of **SOM**s classify documents and assign the dominant term as name for the cluster. For real world large scale collections, one two-dimensional grid is not sufficient. It would be either too big or each node would contain too many documents consequently. Neither would be helpful for users, therefore, a layered architecture is adopted. The highest layer consists of nodes which represent clusters of documents. The documents of these nodes are again analyzed by a **SOM**. For the user, the system consists of several two-dimensional maps of terms where similar terms are close to each other. After choosing one node, he may reach another two-dimensional **SOM**.

The **information retrieval** paradigm for the **SOM** is browsing and navigating between layers of maps. The SOM seems to be a very natural visualization. However, the **SOM** approach has some serious drawbacks.

- The interface for interacting with several layers of maps makes the system difficult to browse.
- Users of large text collections need primarily search mechanisms which the SOM itself does not offer.
- The similarity of the document collection is reduced to two dimensions omitting many potentially interesting aspects.
- The SOM unfolds its advantages for human-computer-interaction better for a small number of documents. A very encouraging application would be the clustering of the result set. The neurons would fit on one screen, the number of terms would be limited and therefore, the reduction to two dimensions would not omit so many aspects.

User Classification and Personalization

Adaptive **information retrieval** approaches intend to tailor the results of a system to one user and his interests and preferences. The most popular **representation** scheme relies on the **representation** scheme used in **information retrieval** where a document-term-matrix stores the importance or weight of each term for each document. When a term appears in a document, this weight should be different form zero. User interest can

also be stored like a document. Then the interest is a vector of terms. These terms can be ones that a user has entered or selected in a user interface or which the system has extracted from documents for which the user has shown interest by viewing or downloading them (Agichtein et al., 2006).

An example for such a system is UCAIR which can be installed as a browser plugin. UCAIR relies on a standard web search engine to obtain a search result and a primary ranking. This **ranking** is now being modified by re-**ranking** the documents based on implicit feedback and a stored user interest profile (Shen et al., 2005).

Most systems use this method of storing the user interest in a term vector. However, this method has several drawbacks. The interest profile may not be stable and the user may have a variety of diverging interests for work and leisure which are mixed in one profile.

Advanced **individualization** techniques personalize the underlying system functions. The results of empirical studies have shown that relevance feedback is an effective technique to improve retrieval quality. Learning methods for **information retrieval** need to extend the range of relevance feedback effects beyond the modification of the query in order to achieve long-term adaptation to the subjective point of view of the user. The mere change of the query often results in improved quality; however, the information is lost after the current session.

Some systems change the document **representation** according to the relevance feedback information. In a vector space metaphor, the relevant documents are moved toward the query **representation**. This approach also comprises some problems. Because only a fraction of the documents are affected by the modifications, the basic data from the **indexing** process is changed to a somewhat heterogeneous state. The original **indexing** result is not available anymore.

Certainly, this technique is inadequate for fusion approaches where several retrieval methods are combined. In this case, several basic **representation**s would need to be changed according to the influence of the corresponding methods on the relevant documents. The indexes are usually heterogeneous, which is often considered an advantage of fusion approaches. A high computational overload would be the consequence.

The MIMOR (Multiple Indexing and Method-Object Relations) approach does not rely on changes to the document or the query **representation** when processing

relevance feedback information for personalization. Instead, it focuses on the central aspect of a retrieval function, the calculation of the similarity between document and query. Like other fusion methods, MIMOR accepts the result of individual retrieval systems like from a black box. These results are fused by a linear combination which is stored during many sessions. The weights for the systems experience a change through learning. They adapt according to relevance feedback information provided by users and create a long-term model for future use. That way, MIMOR learns which systems were successful in the past (Mandl & Womser-Hacker, 2004).

FUTURE TRENDS

Information retrieval systems are applied in more and more complex and diverse environments. Searching e-mail, social computing collections and other specific domains pose new challenges which lead to innovative systems. These retrieval applications require thorough and user oriented evaluation. New evaluation measures and standardized test collections are necessary to achieve reliable evaluation results.

In user adaptation, recommendation systems are an important trend for future improvement. Recommendation systems need to be seen in the context of social computing applications. System developers face the growth of user generated content which allows new reasoning methods.

New application like question answering relying on more intelligent processing can be expected to gain more market share in the near future (Hartrumpf, 2006)

CONCLUSION

Knowledge management is of main importance for the information society. Documents written in natural language contain an important share of the knowledge available. Consequently, retrieval is crucial for the success of knowledge management systems. AI technologies have been widely applied in retrieval systems. Exploiting knowledge more efficiently is a major research field. In addition, user oriented value added systems require intelligent processing and machine learning in many forms.

An important future trend for AI methods in IR will be the context specific adaptation of retrieval methods.

Machine learning can be applied to find optimized functions for collections or queries.

REFERENCES

Agichtein, E., Brill, E., Dumais, S., & Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. *Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR) Seattle. ACM Press. 3-10.

de Almeida, H.M., Gonçalves, M.A, Cristo, M., & Calado, P. (2007). A combined component approach for finding collection-adapted ranking functions based on genetic programming. *Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR) Amsterdam. ACM Press. 399-406.

Belkin, R. (2000). Finding out about: a Cognitive Perspective on Search Engine Technology and the WWW. Cambridge et al.: Cambridge University Press.

Brusilovsky, P., Kobsa, A. & Nejdl, W. (eds.) (2007). *The adaptive Web: Methods and strategies of Web personalization.* Heidelberg: Springer.

Boughanem, M., & Soulé-Dupuy, C. (1998). Mercure at trec6. In Voorhees, E. & Harman, D. (eds.). *The sixth text retrieval conf (TREC-6).* NIST Special Publ 500-240. Gaithersburg, MY.

Borodin, A., Roberts, G., Rosenthal, J. & Tsaparas, P. (2005). Link analysis ranking: algorithms, theory, and experiments. *ACM Transactions on Internet Technology* (TOIT) 5(1), 231-297.

Hartrumpf, S. (2006). Extending Knowledge and Deepening Linguistic Processing for the Question Answering System InSicht. Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evalution Forum, CLEF. [LNCS 4022] Springer. 361-369

Ivory M. & Hearst, M. (2002). Statistical Profiles of Highly-Rated Sites. *ACM CHI Conference on Human Factors in Computing Systems*. ACM Press. 367-374.

Joachims, T., & Radlinski, F. (2007). Search engines that learn from implicit feedback. *IEEE Computer 40*(8), 34-40.

Kohonen T. (1998). Self-organization of very large document collections: state of the art. In *Proceedings 8th Intl Conf Artificial Neural Networks*. Springer, 1. 65-74.

Kwok, K., & Grunfeld, L. (1996). TREC-4 ad-hoc, routing retrieval and filtering experiments using PIRCS. In: Harman Donna (ed.). The fourth Text Retrieval Conference (TREC-4). NIST Special Publ 500-236. Gaithersburg, MY.

Lin, J., & Demner-Fushman, D. (2006). The role of knowledge in conceptual retrieval: a study in the domain of clinical medicine. In *Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR) Seattle. ACM Press. 99-106

Mandl, T. (2000). Tolerant Information Retrieval with Backpropagation Networks. *Neural Computing & Applications* 9(4), 280-289.

Mandl, T. (2006). Implementation and evaluation of a quality based search engine. In *Proceedings of the 17th ACM Conference on Hypertext and Hypermedia* (HT '06) Odense, Denmark. ACM Press. 73-84.

Mandl, T., & Womser-Hacker, C. (2004). A Framework for long-term Learning of Topical User Preferences in Information Retrieval. *New Library World, 105*(5/6) 184-195.

Savoy, J. (2003). Cross-language information retrieval: experiments based on CLEF 2000 corpora Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys 34*(1), 1-47.

Shen, X., Tan, B. & Zhai, C. (2005). Context-sensitive information retrieval using implicit feedback. *Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR), Seattle, WA. ACM Press. 43-50.

Singh, S., & Dey Lipika (2005). A rough-fuzzy document grading system for customized text information retrieval. *Information Processing & Management 41*(2), 195-216.

Zhang, D., Chen, X. & Lee, W. (2005). Text classification with kernels on the multinomial manifold. *Annual International ACM Conference on Research and Development in Information Retrieval* (SIGIR) Seattle. ACM Press. 266-273.

A

KEY TERMS

Adaptation: Adaptation is a process of modification based on input or observation. An information system should adapt itself to the specific needs of individual users in order to produce optimized results.

Indexing: Indexing means the assignment of terms (words) which represent a document in an index. Indexing can be carried out manually or automatically. Automatic indexing requires the elimination of stop-words and stemming.

Information Retrieval: Information retrieval is concerned with the representation and knowledge and subsequent search for relevant information within these knowledge sources. Information retrieval provides the technology behind search engines.

Link Analysis: The links between pages on the web are a large knowledge source which is exploited by link analysis algorithms for many ends. Many algorithms similar to PageRank determine a quality or authority score based on the number of in-coming links of a page. Furthermore, link analysis is applied to identify thematically similar pages, web communities and other social structures.

Recommendation Systems: Actions or content is suggested to the user based on past experience collected from other users. Very often, documents are recommended based on similarity profiles between users.

Term Expansion: Terms not present in the original query to an information retrieval system entered by the user are added automatically. The expanded query is then sent to the system again.

Weighting: Weighting determines the importance of a term for a document. Weights are calculated using many different formulas which consider the frequency of each term in a document and in the collection as well as the length of the document and the average or maximum length of any document in the collection.

Artificial Intelligence in Computer–Aided Diagnosis

Paulo Eduardo Ambrósio
Santa Cruz State University, Brazil

INTRODUCTION

Professionals of the medical radiology area depend directly on the process of decision making in their daily activities. This process is mainly based on the analysis of a great amount of information obtained for the evaluation of radiographic images.

Some studies demonstrate the great capacity of Artificial Neural Networks (ANN) in support systems for diagnosis, mainly in applications as pattern classification.

The objective of this article is to present the development of an ANN-based system, verifying its behavior as a feature extraction and dimensionality reduction tool, for recognition and characterization of patterns, for posterior classification in normal and abnormal patterns.

BACKGROUND

The computer-aided diagnosis (CAD) is considered one of the main areas of research of the medical images and radiological diagnosis (Doi, 2005).

According to Giger (2002) "In the future, is probable that all the medical images have some form of executed CAD to benefit to the results and the patient cares".

The diagnosis of the radiologist is normally based on qualitative interpretation of the analyzed data, that can be influenced and be harmed by many factors, as low quality of the image, visual fatigue, distraction, overlapping of structures, amongst others (Azevedo-Marques, 2001). Moreover, the human beings possess limitations in its visual ability, which can harm the analysis of a medical image, mainly in the detection of determined presented patterns (Giger, 2002).

Research demonstrates that when the analysis is carried out by two radiologists, the diagnosis sensitivity is significantly increases (Thurfjell *et al.*, 1994). In this direction, the CAD can be used as a second specialist,

when providing the computer reply as a second opinion (Doi, 2005).

Many works analyze the radiologist performance front the use of a CAD systems, of which we detach the research of Jiang *et al.* (2001) and Fenton *et al.* (2007).

In the development of CAD systems, techniques from two computational areas are normally used: Computer Vision and Artificial Intelligence.

From the area of Computer Vision, techniques of image processing for enhancement, segmentation and feature extraction are used (Azevedo-Marques, 2001).

The enhancement objectives to improve an image to make it more appropriate for a specific application (Gonzalez & Woods, 2001). In applications with digital medical images, the enhancement is important to facilitate the visual analysis on the part of the specialist.

The segmentation is the stage where the image is subdivided in parts or constituent objects (Gonzalez & Woods, 2001). The result of the segmentation is a set of objects that can be analyzed and quantified individually, representing determined characteristic of the original image.

The final stage involved in image processing is the feature extraction, that it basically involves the quantification of elements that compose segmented objects of the original image, such as size, contrast and form.

After concluded this first part, the quantified attributes are used for the classification of the structures identified in the image, normally using methods of Artificial Intelligence. According to Kononenko (2001), the use of Artificial Intelligence in the support to the diagnosis is efficient, for allowing a complex data analysis of simple and direct form.

Many methods and techniques of Artificial Intelligence can be applied in this stage, normally with the objective to identify and to separate the patterns in distinct groups (Theodorides & Koutroumbas, 2003), for example, normal and abnormal patterns. According to Kahn Jr (1994), among the main techniques, can be

cited: rule-based reasoning, artificial neural networks, bayesian networks, case-based reasoning. To these, the statistical methods, the genetic algorithms and the decision trees can be added.

A problem that reaches most of the applications of pattern recognition is the data dimensionality. The dimensionality is associated with the number of attributes that represent a pattern, that is, the dimension of the search space. When this space contains only the most relevant attributes, the classification process is faster and consumes little processing resources (Jain *et al.*, 2000), and also allows for greater precision of the classifier.

In the problems of medical image processing, the importance of the dimensionality reduction is accentuated; therefore normally the images to be processed are composed of a very great number of pixels, used as basic attributes in the classification.

The feature extraction is a common boarding to effect the dimensionality reduction. Of general form, an extraction algorithm creates a new set of attributes from transformations or combinations of the original set.

Some methods are studied with the intention to promote the feature extraction and, consequently, the dimensionality reduction, such as statistical methods, methods based on the signal theory, and artificial neural networks (Verikas & Bacauskiene, 2002).

As example of the use of artificial neural networks in the support to the medical diagnosis, we can cite the research of Papadopoulos *et al.* (2005) and André & Rangayan (2006).

MAIN FOCUS OF THE ARTICLE

In this paper, we also present a proposal of use of Artificial Intelligence in the stage of feature extraction, substituting the traditional techniques of image processing.

Traditionally, the feature extraction is carried out on the basis of statistical or spectral techniques, which result in, for example, texture or geometric attributes.

After these attributes are obtained, techniques of Artificial Intelligence are applied in the pattern classification.

Our proposal is the use of ANN also for feature extraction.

Feature Extraction with ANNs

The feature extraction with the use of Artificial Neural Networks functions basically as a selection of characteristics that represent the original data set.

This selection of characteristics is related to a process in which a data set is transformed into a space of characteristics that, in theory, accurately describes the same information as the original space of the data. However, the transformation is projected in such a way that the data set is represented by a reduced effective characteristic, keeping most of the intrinsic information to the data, that is, the original data set suffers a significant dimensionality reduction (Haykin, 1999).

The dimensionality reduction is extremely useful in applications that involve digital image processing, which normally depend on a very high number of data points to be manipulated.

In summary, the feature extraction with ANNs transforms the original set of pixels into a map, of reduced dimensions, that represents the original image without a significant loss of information.

For this function, self-organizing neural networks are normally used, as for example, the Kohonen's Self-Organizing Map (SOM).

The self-organizing map searches ways to transform one determined pattern into a bi-dimensional map, following a certain topological order (Haykin, 1999). The elements that compose the map are distributed in an only layer, having formed a grid (Figure 1).

Figure 1. Illustrative representation of a Kohonen's self-organizing map
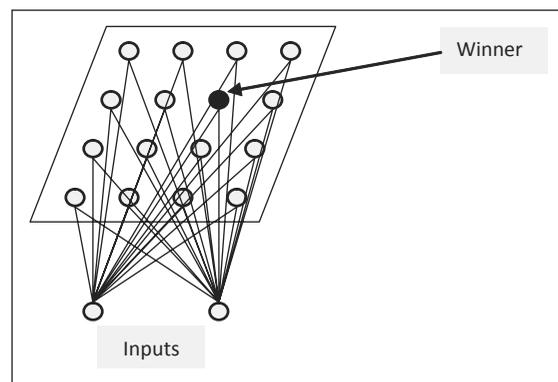
All the elements of the grid receive the input signal of all variables, associated to its respective weights. The calculation of its value of exit is carried through by one determined function, on the basis of the weights of the connections, and it is used to identify the winning element.

Mathematically, each element of the grid is represented by a vector composed of the weights of connection, with the same dimension of the input space, that is, the amount of elements that compose the vector corresponds to the amount of input variables of the problem (Haykin, 1999).

Methodology

As application example, a self-organizing neural network for the feature extraction of images of chest radiographs was developed, objectifying the characterization of normal and abnormal patterns.

Each original image was divided in 12 parts, having as base the anatomical division normally used in the diagnosis of the radiologist. Each part is formed by approximately 250,000 pixels.

With the use of the proposal self-organizing network, a reduction for only 240 representative elements was obtained, with satisfactory results in the final pattern classification.

A detailed description of the methodology can be found in (Ambrósio, 2007; Azevedo-Marques *et al.*, 2007).

FUTURE TRENDS

The developed study shows the possibilities of application of the self-organizing networks in the feature extraction and dimensionality reduction; however, other types of neural networks can also be used for this purpose. New studies need to be carried out to compare the results and adequacy of the methodology.

CONCLUSION

The contribution of the Information Technology is undeniable as support tool to the medical decision making. The Artificial Intelligence presents itself as a great source of important techniques to be used in this direction.

It can be evidenced that the technique of artificial neural networks highlights its great versatility and robustness, providing sufficiently satisfactory results, when used and implemented well.

The use of an automatic system of image analysis can assist the radiologist, when used as a tool of 'second opinion', or second reading, in the analysis of possible inexact cases.

It is also observed that the use of the proposed methodology represents a significant profit in the image processing of chest radiographs, for its peculiar characteristics.

REFERENCES

Ambrósio, P.E. (2007). *Self-Organizing Neural Networks in the Characterization of Interstitial Lung Diseases in Chest Radiographs.* Thesis. Ribeirão Preto: School of Medicine of Ribeirao Preto, University of Sao Paulo. [*In Portuguese*]

André, T. C. S. S.; Rangayan, R. M. (2006). Classification of breast masses in mammograms using neural networks with shape, edge-sharpness and texture features. *Journal of Electronic Imaging*. 15(1).

Azevedo-Marques, P.M. (2001). Diagnóstico Auxiliado por Computador na Radiologia. *Radiologia Brasileira.* 34(5), 285-293. [*In Portuguese*]

Azevedo-Marques, P.M., Ambrósio, P.E., Pereira-Junior, R.R., Valini, R.A. & Salomão, S.C. (2007). Characterization of Interstitial Lung Disease in Chest Radiographs using SOM Artificial Neural Network. *International Journal of Computer Assisted Radiology and Surgery. 2* (suppl. 1), 368-370.

Doi, K. (2005). Current Status and Future Potential of Computer-Aided Diagnosis in Medical Imaging. *The British Journal of Radiology.* 78(special issue), S3-S19.

Fenton, J. J.; Taplin, S. H.; Carney, P. A.; Abraham, L.; Sickles, E. A.; D'Orsi, C.; Berns, E. A.; Cutter, G.; Hendrick, R. E.; Barlow, W. E.; Elmore, J. G. (2007). Influence of computer-aided detection on performance of screening mammography. *New England Journal of Medicine*. 356 (14), 1399-1409.

Giger, M.L. (2002). Computer-Aided Diagnosis in Radiology. *Academic Radiology.* 9, 1-3.

Gonzalez, R.C. & Woods, R.E. (2001). *Digital Image Processing* (2nd ed.). Reading, MA: Addison-Wesley.

Haykin, S. (1999). *Neural Networks* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Jain, A.K., Duin, R.P.W. & Mao, J. (2000). Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 22(1), 4-37.

Jiang, Y.; Nishikawa, R. M.; Schimidt, R. A.; Toledano, A. Y.; Doi, K. (2001) Potential of computer-aided diagnosis to reduce variability in radiologists' interpretations of mammograms depicting microcalcification. *Radiology*. 220(3), 787-794.

Kahn-Jr, C.E. (1994). Artificial Intelligence in Radiology: Decision Support Systems. *RadioGraphics.* 14(4), 849-861.

Kononenko, I. (2001). Machine Learning for Medical Diagnosis: History, State of the Art and Perspective. *Artificial Intelligence in Medicine*, 23, 89-109.

Papadopoulos, A; Fotiadis, D. I.; Likas, A. (2005). Characterization of clustered microcalcifications in digitized mammograms using neural networks and support vector machines. *Artificial Intelligence in Medicine*. 34(2), 141-150.

Theodoridis, S. & Koutroumbas, K. (2003). *Pattern Recognition* (2nd ed.). Amsterdam: Elsevier.

Thurfjell, E.L., Lernevall, K.A. & Taube, A.A.S. (1994). Benefit of Independent Double Reading in a Population-Based Mammography Screening Program. *Radiology*. 191, 241-244.

Verikas, A. & Bacauskienes, M. (2003). Feature Selection with Neural Networks. *Pattern Recognition Letters.* 23(11), 1323-1335.

KEY TERMS

Computer-Aided Diagnosis: Research area that enclose the development of computational techniques and procedures for aid to the health professionals in process of decision making for the medical diagnosis.

Dimensionality Reduction: Finding a reduced data set, with the capacity of mapping a bigger set.

Feature Extraction: Finding of representative features of a determined problem from samples with different characteristics.

Medical Images: Images generated in special equipment, used for aid to the medical diagnosis. Ex.: X-Ray images, Computer Tomography, Magnetic Resonance Images.

Pattern Recognition: Research area that enclose the development of methods and automatized techniques for identification and classification of samples in specific groups, in accordance with representative characteristics.

Radiological Diagnosis: Medical diagnosis based in analysis and interpretation of patterns observed in medical images.

Self-Organizing Maps: Category of algorithms based on artificial neural networks that searches, by means of self-organization, to create a map of characteristics that represents the involved samples in a determined problem.

Artificial Neural Networks and Cognitive Modelling

Amanda J.C. Sharkey
University of Sheffield, UK

INTRODUCTION

In their heyday, artificial neural networks promised a radically new approach to cognitive modelling. The connectionist approach spawned a number of influential, and controversial, cognitive models. In this article, we consider the main characteristics of the approach, look at the factors leading to its enthusiastic adoption, and discuss the extent to which it differs from earlier computational models. Connectionist cognitive models have made a significant impact on the study of mind. However connectionism is no longer in its prime. Possible reasons for the diminution in its popularity will be identified, together with an attempt to identify its likely future.

The rise of connectionist models dates from the publication in 1986 by Rumelhart and McClelland, of an edited work containing a collection of connectionist models of cognition, each trained by exposure to samples of the required tasks. These volumes set the agenda for connectionist cognitive modellers and offered a methodology that subsequently became the standard. Connectionist cognitive models have since been produced in domains including memory retrieval and category formation, and (in language) phoneme recognition, word recognition, speech perception, acquired dyslexia, language acquisition, and (in vision) edge detection, object and shape recognition. More than twenty years later the impact of this work is still apparent.

BACKGROUND

Seidenberg and McClelland's (1989) model of word pronunciation is a well-known connectionist example. They used backpropagation to train a three-layer network to map an orthographic representation of words and non-words onto a distributed phonological representation, and an orthographic output representation. The model is claimed to provide a good fit to experimental data from human subjects. Humans can make rapid decisions about whether a string of letters is a word or not, (in a lexical decision task), and can readily pronounce both words and non-words. The time they take to do both is affected by a number of factors, including the frequency with which words occur in language, and the regularity of their spelling. The trained artificial neural network outputs both a phonological and an orthographic representation of its input. The phonological representation is taken as the equivalent to pronouncing the word or non-word. The orthographic representation, and the extent to which it duplicates the original input, is taken to be the equivalent of the lexical decision task

The past tense model (McClelland & Rumelhart, 1986) has also been very influential. The model mirrors several aspects of human learning of verb endings. It was trained on examples of the root form of the word as input, and of the past-tense form as output. Each input and output was represented as a set of context-sensitive phonological features, coded and decoded by means of a fixed encoder/decoder network. A goal of the model was to simulate the stage-like sequences of past tense learning shown by humans. Young children first correctly learn the past tense of a few verbs, both regular (e.g. looked) and irregular (e.g. went, or came). In stage 2 they often behave as though they have inferred a general rule for creating the past tense, (adding –ed to the verb stem). But they often over-generalise this rule, and add –ed to irregular verbs (e.g comed). There is a gradual transition to the final stage in which they learn to produce the correct past tense form of both regular and exception words. Thus their performance exhibits a U-shaped function for irregular verbs (initially correct, then often wrong, then correct again).

The model was trained in stages on 506 English verbs. First, it was trained on 10 high frequency verbs (regular, and irregular). Then medium frequency verbs (mostly regular) were introduced and trained for a number of epochs. A dip in performance on the irregular verbs occurred shortly after the introduction

of the medium frequency verbs – a dip followed by a gradual improvement that resembled the U-shaped curve found in human performance.

THE STRENGTHS AND LIMITATIONS OF CONNECTIONIST COGNITIVE MODELLING

The models outlined above exhibit five typical features of connectionist models of cognition: (i) They provide an account that is related to and inspired by the operations of the brain; (ii) They can be used both to model mental processes, and to simulate the actual behaviour involved; (iii) They can provide a 'good fit' to the data from psychology experiments; (iv) The model, and its fit to the data, is achieved without explicit programming and (v) They often provide new accounts of the data. We discuss these features in turn.

First there is the idea that a connectionist cognitive model is inspired by, and related to, the way in which brains work. Connectionism is based on both the alleged operation of the nervous system and on distributed computation. Neuron-like units are connected by means of weighted links, in a manner that resembles the synaptic connections between neurons in the brain. These weighted links capture the knowledge of the system; they may be arrived at either analytically or by "training" the system with repeated presentations of input-output training examples. Much of the interest in connectionist models of cognition was that they offered a new account of the way in which knowledge was represented in the brain. For instance, the behaviour of the past tense learning model can be described in terms of rule following – but its underlying mechanism does not contain any explicit rules. Knowledge about the formation of the past tense is distributed across the weights in the network.

Interest in brain-like computing was fuelled by a growing dissatisfaction with the classical symbolic processing approach to modelling mind and its relationship to the brain. Even though theories of symbol manipulation could account for many aspects of human cognition, there was concern about how such symbols might be learnt and represented in the brain. Functionalism (Putnam, 1975) explicitly insisted that details about how intelligence and reasoning were actually implemented were irrelevant. Concern about the

manipulation of meaningless, ungrounded symbols is exemplified by Searle's Chinese Room thought-experiment (1980). Connectionism, by contrast, offered an approach that was based on learning, made little use of symbols, and was related to the way in which the brain worked. Arguably, one of the main contributions that connectionism has made to the study and understanding of mind has been the development of a shared vocabulary between those interested in cognition, and those interested in studying the brain.

The second and third features relate to the way in which artificial neural nets can both provide a model of a cognitive process and simulate a task, and provide a good fit to the empirical data. In Cognitive Psychology, the emphasis had been on building models that could account for the empirical results from human subjects, but which did not incorporate simulations of experimental tasks. Alternatively, in Artificial Intelligence, models were developed that performed tasks in ways that resembled human behaviour, but which took little account of detailed psychological evidence. However, as in the two models described here, connectionist models both simulated the performance of the human tasks, and were able to fit the data from psychological investigations.

The fourth feature is that of achieving the model and the fit to the data without explicit handwiring. It can be favourably contrasted to the symbolic programming methodology of Artificial Intelligence, where the model is programmed step by step, leaving room for ad hoc modifications and kludges. The fifth characteristic is the possibility of providing a novel explanation of the data. In their model of word pronunciation, Seidenberg and McClelland showed that their artificial neural network provided an integrated (single mechanism) account of data on both regular and exception words where previously the old cognitive modelling conventions had forced an explanation in terms of a dual route. Similarly, the past-tense model was formulated as a challenge to rule-based accounts: although children's performance can be described in terms of rules, it was claimed that the model showed that the same behaviour could be accounted for by means of an underlying mechanism that does not use explicit rules.

In its glory days, connectionism's claims about novel explanations of stimulated much debate. There was also much discussion of the extent to which connectionism could provide an adequate account of higher mental processes. Fodor and Pylyshyn (1988)

mounted an attack on the representational adequacy of connectionism. Connectionists retaliated, and in papers such as van Gelder's (1990) the argument was made that not only could they provide an account of the structure sensitive processes underlying human language, but that connectionism did so in a novel manner: the *eliminative connectionist* position.

Now the dust has subsided, connectionist models do not seem as radically different to other modelling approaches as was once supposed. It was held that one of their strengths was their ability to model mental processes, simulate behaviour, and provide a good fit to data from psychology experiments *without being explicitly programmed to do so*. However, there is now greater awareness that decisions about factors such as the architecture of the net, the form its representations will take, and even the interpretation of its input and output, are tantamount to a form of *indirect, or extensional programming*.

Controlling the content, and presentation of the training sample, is an important aspect of extensional programming. When Pinker and Prince (1988) criticised the past tense model, an important element of their criticisms was that the experimenters had unrealistically tailored the environment to produce the required results, and that the results were an artifact of the training data. Although the results indicated a U-shaped curve in the rate of acquisition, as occurs with children, Pinker and Prince argued that this curve occurred only because the net was exposed to the verbs in an unrealistically structured order. Further research has largely answered these criticisms, but it remains the case that selection of the input, and control of the way that it is presented to the net, affects what the net learns. A similar argument can be made about the selection of input representations.

In summary: there has been debate about the novelty of connectionism, and its ability to account for higher level cognitive processing. There is however general acknowledgement that the approach made a lasting contribution by indicating how cognitive processes could be implemented at the level of neurons. Despite this, the connectionist approach to cognitive modelling is no longer as popular as it once was. Possible reasons are considered below:

- **Difficult challenges:** A possible reason for the diminished popularity of artificial neural nets is that as Elman (2005) suggests, "we have arrived at the point where the easy targets have been identified but the tougher problems remain". Difficult challenges to be met include the idea of scaling up models to account for wider ranges of phenomena, and building models that can account for more than one behaviour.

- **Greater understanding:** As a result of our greater understanding of the operation and inherent limitations of artificial neural nets, some of their attraction has faded with their mystery. They have become part of the arsenal of statistical methods for pattern recognition, and much recent research on artificial neural networks has focused more on questions about whether the best level of generalisation has been efficiently achieved, than on modelling cognition.

Also there is greater knowledge of the limitations of artificial neural nets, such as the problem of the "catastrophic interference" associated with backpropagation. Backpropagation performs impressively when all of the training data are presented to the net on each training cycle, but its results are less impressive when such training is carried out sequentially and a net is fully trained on one set of items before being trained on a new set. The newly learned information often interferes with, and overwrites, previously learned information. For instance, McCloskey and Cohen (1989) used backpropagation to train a net on the arithmetic problem of + 1 addition (e.g. 1+1, 2+1, …, 9+1). They found that when they proceeded to train the same net to add 2 to a given number, it "forgot" how to add 1. Sequential training of this form results in catastrophic interference. Sharkey and Sharkey (1995) demonstrated that it is possible to avoid the problem if the training set is sufficiently representative of the underlying function, or there are enough sequential training sets. In terms of this example, if the function to be learned is both + 1 and + 2, then training sets that incorporate enough examples of each could lead to the net learning to add either 1 or 2 to a given number. However, this is at the expense of being able to discriminate between those items that have been learned from those that have not.

This example is related to another limitation of artificial neural nets: their inability to extrapolate beyond their training set. Although humans can readily grasp the idea of adding one to any given number, it is not so straightforward to train the net to extrapolate beyond the data on which it is trained. It has been argued

(Marcus, 1998) that this inability of artificial neural nets trained using backpropagation to generalise beyond their training space provides a major limitation to the power of connectionist nets: an important one, since humans can readily generalise universal relationships to unfamiliar instances. Clearly there are certain aspects of cognition, particularly those to do with higher level human abilities, such as their reasoning and planning abilities, that are more difficult to capture within connectionist models.

- **Changing zeitgest:** There is now an increased interest in more detailed modelling of brain function, and a concomitant dissatisfaction with the simplicity of cognitive models that often consisted of "a small number of neurons connected in three rows" (Hawkins, 2004). Similarly, there is greater impatience with the emphasis in connectionism on the biologically implausible backpropagation learning algorithm. At the same time, there is greater awareness of the role the body plays in cognition, and the relationships between the body, the brain, and the environment (e.g. Clark, 1999). Traditional connectionist models do not fit easily with the new emphasis on embodied cognition (e.g. Pfeifer and Scheier, 1999).

FUTURE TRENDS

One expected future trend will be to focus on the difficult challenges. It is likely that investigations will explore how models of isolated processes, such as learning the past tense of verbs, might fit into more general accounts of language learning, and of cognition. There are further questions to be addressed about the developmental origin of many aspects of cognition, and the origin and progression of developmental disorders.

Connectionist cognitive modelling is likely to change in response to the new zeitgest. A likely scenario is that artificial neural nets will continue to be used for cognitive modeling, but not exclusively as they were before. They will continue to form part of hybrid approaches to cognition (Sun, 2003), in combination with symbolic methods. Similarly, artificial neural nets can be used in combination with evolutionary algorithms to form the basis of adaptive responses to the environment. Rather than training artificial neural nets, they can be adapted by means of evolutionary methods, and

used as the basis for robotic controllers (e.g. Nolfi and Floreano, 2000). Such changes will ensure a future for connectionist modeling, and stimulate a new set of questions about the emergence of cognition in response to an organism's interaction with the environment.

CONCLUSION

In this article we have described two landmark connectionist cognitive models, and considered their characteristic features. We outlined the debates over the novelty and sufficiency of connectionism for modelling cognition, and argued that in some respects the approach shares features with the modelling approaches that preceded it. Reasons for a gradual waning of interest in connectionism were identified, and possible futures were discussed. Connectionism has had a strong impact on cognitive modelling, and although its relationship to the brain is no longer seen as a strong one, it provided an indication of the way in which cognitive processes could be accounted for in the brain. It is argued here that although the approach is no longer ubiquitous, it will continue to form an important component of future cognitive models, as they take account of the interactions between thought, brains and the environment.

REFERENCES

Clark, A. (1997) *Being there: Putting brain, body and world together again*. Cambridge, MA: MIT Press

Elman, J.L.. (2005) Connectionist models of cognitive development: Where next? *Trends in Cognitive Sciences*, 9, 111-11

Fodor, J.A. & Pylyshyn, Z. (1988) Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28, 3-71.

Hawkins, J. (2004) *On intelligence*. New York: Henry Holt and Company, Owl Books.

Marcus, G.F. (1998) Rethinking eliminative connectionism. *Cognitive Psychology,* 37, 243-282.

McClelland, J.L. & Rumelhart, D.E. & the PDP Research Group *Parallel Distributed Processing Vol 2: Psychological and Biological Models*. Cambridge, MA: MIT Press. (1986)

McCloskey, M. & Cohen, N.J. (1989) Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, Vol 24, 109-165.

Nolfi, S. and Floreano, D. (2004) *Evolutionary robotics: The biology, intelligence and technology of self-organising machines*. Cambridge, MA: MIT Press

Pinker, S. & Prince, A. (1988) On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. In *Connections and symbols* (S., Pinker and J. Mehler, Eds), Cambridge, MA: Bradford/MIT Press pp 73-194

Pfeifer, R. and Scheier, C. (1999) *Understanding intelligence*. Cambridge, MA: MIT Press

Putnam, H. (1975) Philosophy and our mental life. In H. Putnam (Ed.) *Mind, language and reality: Philosophical papers (vol 2)* Cambridge, UK: Cambridge University Press, pp 48-73

Rumelhart, D.E. and McClelland, J.L. (1986) On learning the past tenses of English verbs. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 2. Psychological and Biological Models*. (Rumelhart, D.E. and McClelland, J.L., eds) pp 216-271, MIT Press

Seidenberg, M.S. and McClelland, J.L. (1989) A distributed developmental model of visual word recognition and naming. *Psychological Review*, 96, 523-568.

Searle, J.R. (1980) Minds, brains and programs. *Behavioural and Brain Sciences,* 3: 417-424. Reprinted in J. Haugeland. Ed. *Mind design.* Montgomery, VT: Bradford Books, 1981.

Sharkey, N.E. & Sharkey, A.J.C. (1995) An Analysis of Catastrophic Interference, *Connection Science*, 7, 3/4, 313-341.

Sharkey, A.J.C. and Sharkey, N. (2003) Cognitive Modelling: Psychology and Connectionism. In M.A. Arbib (Ed) *The Handbook of Brain Theory and Neural Networks,* A Bradford Book: MIT Press

Sun, R. (2003) Hybrid Connectionist/Symbolic Systems. In M.A. Arbib (Ed) *The Handbook of Brain Theory and Neural Networks,* A Bradford Book: MIT Press pp 543-547

van Gelder, T. (1990) Compositionality: A Connectionist variation on a classical theme. *Cognitive Science*, 14, pp 355-364.

A

KEY TERMS

Chinese Room: In Searle's thought experiment, he asks us to imagine a man sitting in a room with a number of rule books. A set of symbols is passed into the room. The man processes the symbols according to the rule books, and passes a new set of symbols out of the room. The symbols posted into the room correspond to a Chinese question, and the symbols he passes out are the answer to the question, in Chinese. However, the man following the rules has no knowledge of Chinese. The example suggests a computer program could similarly follow rules in order to answer a question without any understanding.

Classical Symbol Processing: The classical view of cognition was that it was analogous to symbolic computation in digital computers. Information is represented as strings of symbols, and cognitive processing involves the manipulation of these strings by means of a set of rules. Under this view, the details of how such computation is implemented are not considered important.

Connectionism: Connectionism is the term used to describe the application of artificial neural networks to the study of mind. In connectionist accounts, knowledge is represented in the strength of connections between a set of artificial neurons.

Eliminative Connectionism: The eliminative connectionist is concerned to provide an account of cognition that eschews symbols, and operates at the subsymbolic level. For instance, the concept of "dog" could be captured in a distributed representation as a number of input features (e.g. four-footed, furry, barks etc) and would then exist in the net in the form of the weighted links between its neuron like units.

Generalisation: Artificial neural networks, once trained, are able to generalise beyond the items on which they were trained and to produce a similar output in response to inputs that are similar to those encountered in training

Implementational Connectionism: In this less extreme version of connectionism, the goal is to find

a means of implementing classical symbol processing using artificial networks – and to find a way of accounting for symbol processing at the level of neurons.

Lexical Decision: Lexical decision tasks are a measure devised to look at the processes involved in word recognition. A word or pseudoword (a meaningless string of letters, conforming to spelling rules) is presented, and the reader is asked to press a button to indicate whether the display was a word or not. The time taken to make the decision is recorded in milliseconds. The measure can provide an indication of various aspects of word processing – for instance how familiar the word is to the reader.

Artificial NeuroGlial Networks

A

Ana Belén Porto Pazos
University of A Coruña, Spain

Alberto Alvarellos González
University of A Coruña, Spain

Félix Montañés Pazos
University of A Coruña, Spain

INTRODUCTION

More than 50 years ago **connectionist systems** (**CSs**) were created with the purpose to process information in the computers like the human brain (McCulloch & Pitts, 1943). Since that time these systems have advanced considerably and nowadays they allow us to resolve complex problems in many disciplines (classification, clustering, regression, etc.). But this advance is not enough. There are still a lot of limitations when these systems are used (Dorado, 1999). Mostly the improvements were obtained following two different ways. Many researchers have preferred the construction of **artificial neural networks** (**ANNs**) based in mathematic models with diverse equations which lead its functioning (Cortes & Vapnik, 1995; Haykin, 1999). Otherwise other researchers have pretended the most possibly to make alike these systems to human brain (Rabuñal, 1999; Porto, 2004).

The systems included in this article have emerged following the second way of investigation. **CSs** which pretend to imitate the neuroglial nets of the brain are introduced. These systems are named **Artificial NeuroGlial Networks** (**ANGNs**) (Porto, 2004). These **CSs** are not only made of neuron, but also from elements which imitate glial neurons named **astrocytes** (Araque, 1999). These systems, which have hybrid training, have demonstrated efficacy when resolving classification problems with totally connected feed-forward multilayer networks, without backpropagation and lateral connections.

BACKGROUND

The **ANNs** or CSs emulate the biological neural networks in that they do not require the programming of tasks but generalise and learn from experience. Current **ANNs** are composed by a set of very simple processing elements (PEs) that emulate the biological neurons and by a certain number of connections between them.

Until now, researchers that pretend to emulate the brain, have tried to represent in ANNs the importance the neurons have in the Nervous System (NS). However, during the last decades research has advanced remarkably in the Neuroscience field, and increasingly complex neural circuits, as well as the Glial System (GS), are being observed closely. The importance of the functions of the GS leads researchers to think that their participation in the processing of information in the NS is much more relevant than previously assumed. In that case, it may be useful to integrate into the artificial models other elements that are not neurons.

Since the late 80s, the application of innovative and carefully developed cellular and physiological techniques (such as patch-clamp, fluorescent ion-sensible images, confocal microscopy and molecular biology) to glial studies has defied the classic idea that **astrocytes** merely provide a structural and trophic support to neurons and suggests that these elements play more active roles in the physiology of the Central Nervous System.

New discoveries are now unveiling that the glia is intimately linked to the active control of neural activity and takes part in the regulation of synaptic neurotransmission (Perea & Araque, 2007). Abundant evidence has suggested the existence of bidirectional communication between astrocytes and neurons, and the important active role of the astrocytes in the NS's

physiology (Araque et al., 2001; Perea & Araque, 2005). This evidence has led to the proposal of a new concept in synaptic physiology, the tripartite synapse, which consists of three functional elements: the presynaptic and postsynaptic elements and the surrounding astrocytes (Araque et al., 1999). The communication between these three elements has highly complex characteristics, which seem to reflect more reliably the complexity of the **information processing** between the elements of the NS (Martin & Araque, 2005).

So there is no question about the existence of communication between astrocytes and neurons (Perea & Araque, 2002). In order to understand the motives of this reciprocated signalling, we must know the differences and similarities that exist between their properties. Only a decade ago, it would have been absurd to suggest that these two cell types have very similar functions; now we realise that the similarities are striking from the perspective of chemical signalling. Both cell types receive chemical inputs that have an impact on the ionotropic and metabotropic receptors. Following this integration, both cell types send signals to their neighbours through the release of chemical transmittors. Both the neuron-to-neuron signalling and the neuron-to-astrocyte signalling show plastic properties that depend on the activity (Pasti et al., 1997). The main difference between **astrocytes** and neurons is that many neurons extend their axons over large distances and conduct action potentials of short duration at high speed, whereas the **astrocytes** do not exhibit any electric excitability but conduct calcium spikes of long duration (tens of seconds) over short distances and at low speed. The fast signalling, and the input/output functions in the central NS that require speed, seem to belong to the neural domain. But what happens with slower events, such as the induction of memories, and other abstract processes such as thought processes? Does the signalling between astrocytes contribute to their control? As long as there is no answer to these questions, research must continue; the present work offers new ways to advance through the use of Artificial Intelligence (AI) techniques.

Therefore not only it is pretended to improve the **CSs** incorporating elements imitating astrocytes, but it is also intended to benefit Neuroscience with the study of brain circuits since other point of view, the AI.

The most recent works in this area are presented by Porto et al (Porto et al., 2007; Porto et al., 2005; Porto, 2004).

MAIN FOCUS OF THE ARTICLE

All the design possibilities, for the architecture as well as for the training process of an ANN, are basically oriented towards minimising the error level or reducing the system's learning time. As such, it is in the optimisation process of a mechanism, in case the ANN, that we must find the solution for the many parameters of the elements and the connections between them.

Considering possible future improvements that optimize an ANN with respect to minimal error and minimal training time, our models will be the brain circuits, in which the participation of elements of the GS is crucial to process the information. In order to design the integration of these elements into the ANN and elaborate a learning method for the resulting ANGN that allows us to check whether there is an improvement in these systems, we have analysed the main existing training methods that will be used for the elaboration. We have analysed Non-Supervised and Supervised Training methods, and other methods that use or combine some of their characteristics and complete the analysis: Training by Reinforcement, Hybrid Training and Evolutionary Training.

Observed Limitations

Several experiments with ANNs have shown the existence of conflicts between the functioning of the CS and biological neuron networks, due to the use of methods that did not reflect reality. For instance, in the case of a multilayer perceptron, which is a simple CS, the synaptic connections between the PEs have weights that can be excitatory or inhibitory, whereas in the natural NS, are the neurons that seem to represent these functions, not the connections; recent research (Perea & Araque, 2007) indicates that the cells of the GS, more concretely the astrocytes, also play an important role.

Another limitation concerns the learning algorithm known as "Backpropagation", which implies that the change of the connections value requires the backwards transmission of the error signal in the ANN. It was traditionally assumed that this behaviour was impossible in a natural neuron, which, according to the "dynamic polarisation" theory of Ramón y Cajal (1911), is unable to efficiently transmit information inversely through the axon until reaching the cellular soma; new research however has discovered that neurons can send information to presynaptic neurons

A

under certain conditions, either by means of existing mechanisms in the dendrites or else through various interventions of glial cells such as astrocytes.

If the learning is supervised, it implies the existence of an "instructor", which in the context of the brain means a set of neurons that behave differently from the rest in order to guide the process. At present, the existence of this type of neurons is biologically indemonstrable, but the GS seems to be strongly implied in this orientation and may be the element that configures an instructor that until now had not been considered.

It is in this context that the present study analyses to what extent the latest discoveries in Neuroscience (Araque et al., 2001; Perea & Araque, 2002) contribute to these networks: discoveries that proceed from cerebral activity in areas that are believed to be involved in the learning and processing of information (Porto et al., 2007).

Artificial Neuroglial Networks

Many researchers have used the current potential of computers and the efficiency of computational models to elaborate "biological" computational models and reach a better understanding of the structure and behaviour of both pyramidal neurons, which are believed to be involved in learning and memory processes (LeRay et al., 2004, Fernández et al., 2007), and astrocytes (Porto, 2004; Perea & Araque, 2002). These models have provided a better understanding of the causes and factors that are involved in the specific functioning of biological circuits. The present work will use these new insights to progress in the field of Computer Sciences and more concretely in AI.

Figure 1. Artificial NeuroGlial network scheme

We present ANGNs (figure 1) that include both artificial neurons and processing control elements that represent the astrocytes, and whose functioning follows the steps that were successfully applied in the construction and use of CS: design, training, testing and execution.

Also, since the computational studies of the learning with **ANNs** are beginning to converge towards evolutionary computation methods (Dorado, 1999), we will combine the optimisation in the modification of the weights (according to the results of the biological models) with the use of Genetic Algorithms (GAs) in order to find the best solution for a given problem. This evolutionary technique was found to be very efficient in the training phase of the CS (Rabuñal, 1998), because it helps to adapt the CS to the optimal solution according to the inputs that enter the system and the outputs that must be produced by the system. This adaptation phenomenon takes place in the brain thanks to the plasticity of its elements and may be partly controlled by the GS; it is for this reason that we consider the GA as a part of the "artificial glia". The result of this combination is a hybrid learning method (Porto, 2004).

The design of the **ANGNs** is oriented towards classification problems that are solved by means of simple networks, i.e. multilayer networks, although future research may lead to the design of models in more complex networks. It seems a logical approach to start the design of these new models with simple ANNs, and to orientate the latest discoveries on **astrocytes** and pyramidal neurons in information processing towards their use in classification networks, since the control of the reinforcement or weakening of the connections in the brain is related to the adaptation or plasticity of the connections, which lead to the generation of activation ways. This process can therefore improve the classification of the patterns and their recognition by the **ANGN**.

A detailed description of the functioning of the **ANGNs** and results with these systems can be found in Porto et al (Porto, 2004; Porto et al., 2005; Porto et al., 2007).

FUTURE TRENDS

We keep on analysing other synaptic modification possibilities based on brain behaviour to apply them
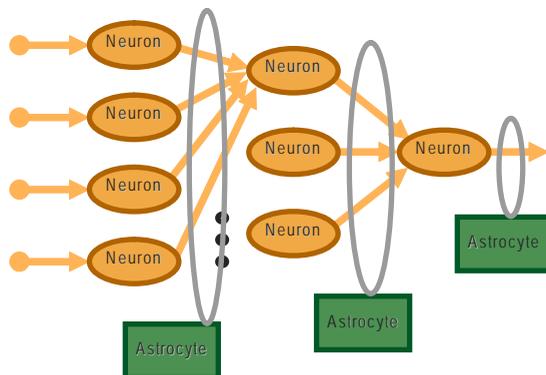
to new **CSs** which can solve simple problems with simple architectures.

Moreover, given that it has been proved that the glia acts upon complex brain circuits, and that the more an individual's brain has developed, the more glia he has in his nervous system (following what Cajal said one hundred years ago (Ramón y Cajal, 1911), we are applying the observed brain behaviour to more complex network architectures. Particularly after having checked that a more complex network architecture achieved better results in the problem presented here.

For the same reason, we intend to analyse how the new **CSs** solve complex problems, for instance time processing ones where totally or partially recurrent networks would play a role. These networks could combine their functioning with this new behaviour.

CONCLUSION

This article presents CSs composed by artificial neurons and artificial glial cells. The design of artificial models did not aim at obtaining a perfect copy of the natural model but a series of behaviours whose final functioning is approached to it as much as possible. Nevertheless, a close similarity between both is indispensable to improve the output, and may result in more "intelligent" behaviours.

The synaptic modifications introduced in the CSs, and based on the modelled brain processes enhance the training of multilayer architectures.

We must remember that the innovation of the existing ANNs models towards the development of new architectures is conditioned by the need to integrate the new parameters into the learning algorithms so that they can adjust their values. New parameters, that provide the process element models of the ANNs with new functionalities, are harder to come by than optimizations of the most frequently used algorithms that increase the calculations and basically work on the computational side of the algorithm. The ANGNs integrate new elements and thanks to a hybrid method this approach did not complicate the training process.

The research with these ANGNs benefits AI because it can improve **information processing** capabilities which would allow us to deal with a wider range of problems. Moreover, this has indirectly benefited Neuroscience since experiments with computational models that simulate brain circuits pave the way for difficult experiments carried out in laboratories, as well as providing new ideas for research.

REFERENCES

Araque, A., Púrpura, V., Sanzgiri, R., & Haydon, P. G. (1999). Tripartite synapses: glia, the unacknowledged partner. Trends in Neuroscience, 22(5).

Araque, A., Carmignoto, G., & Haydon, P. G. (2001): Dynamic Signaling Between Astrocytes and Neurons. Annu. Rev. Physiol, 63, 795-813.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20.

Dorado, J. (1999). Modelo de un Sistema para la Selección Automática en Dominios Complejos, con una Estrategia Cooperativa, de Conjuntos de Entrenamiento y Arquitecturas Ideales de Redes de Neuronas Artificiales Utilizando Algoritmos Genéticos. Tesis Doctoral. Facultad de Informática. Universidade da Coruña.

Fernández, D., Fuenzalida, M., Porto, A., & Buño, W. (2007). Selective shunting of NMDA EPSP component by the slow after hyperpolarization regulates synaptic integration at apical dendrites of CA1 pyramidal neurons. Journal of Neurophysiology, 97, 3242-3255.

Haykin, S. (1999). Neural Networks. 2nd Edition, Prentice Hall.

LeRay, D., Fernández, D., Porto, A., Fuenzalida, M., & Buño, W. (2004). Heterosynaptic Metaplastic Regulation of Synaptic Efficacy in CA1 Pyramidal Neurons of Rat Hippocampus. Hippocampus 14, 1011-1025.

Martín, E.D., & Araque, A. (2005). Astrocytes and The Biological Neural Networks. Artificial Neural Networks in Real-Life Applications. Hershey PA, USA: Idea Group Inc.

McCulloch, W.S., & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 5, 115-133.

Pasti, L., Volterra, A., Pozzan, R., & Carmignoto, G. (1997). Intracellular calcium oscillations in astrocytes: a highly plastic, bidirectional form of communication between neurons and astrocytes in situ. Journal of Neuroscience, 17, 7817-7830.

Perea, G., & Araque, A. (2002). Communication betwen astrocytes and neurons: a complex language. Journal of Physiology. Paris: Elsevier Science.

Perea, G., & Araque, A. (2005). Properties of synaptically evoked astrocyte calcium signal reveal synaptic information processing by astrocytes. The Journal of Neuroscience, 25(9), 2192-2203.

Perea, G., & Araque, A. (2007). Astrocytes Potentiate Transmitter Release at Single Hippocampal Synapses. Science, 317(5841), 1083 – 1086.

Porto, A. (2004). Computational Models for optimizing the Learning and the Information Processing in Adaptive Systems, Ph.D. Thesis, Faculty of Computer Science, University of A Coruña.

Porto, A., Araque, A., & Pazos, A. (2005). Artificial Neural Networks based on Brain Circuits Behaviour and Genetic Algorithms. LNCS. 3512, 99-106.

Porto, A., Araque, A., Rabuñal, J., Dorado, J., & Pazos, A. (2007). A New Hybrid Evolutionary Mechanism Based on Unsupervised Learning for Connectionist Systems. Neurocomputing, 70(16-18), 2799-2808.

Rabuñal, J. (1998). Entrenamiento de Redes de Neuronas Artificiales con Algoritmos Genéticos. Tesis de Licenciatura. Dep. Computación. Facultad de Informática. Universidade da Coruña.

Ramón y Cajal, S. (1911). Histologie du sisteme nerveux de l`homme et des vertebres. Maloine, Paris.

KEY TERMS

Artificial Neural Network: A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Astrocytes: Astrocytes are a sub-type of the glial cells in the brain. They perform many functions, including the formation of the blood-brain barrier, the provision of nutrients to the nervous tissue, and play a principal role in the repair and scarring process in the brain. They modulate the synaptic transmission and recently their crucial role in the information processing was discovered.

Backpropagation Algorithm: A supervised learning technique used for training **ANNs**, based on minimising the error obtained from the comparison between the outputs that the network gives after the application of a set of network inputs and the outputs it should give (the desired outputs).

Evolutionary Computation: Solution approach guided by biological evolution, which begins with potential solution models, then iteratively applies algorithms to find the fittest models from the set to serve as inputs to the next iteration, ultimately leading to a model that best represents the data.

Genetic Algorithms: Genetic algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

Glial Sytem: Commonly called glia (greek for "glue"), are non-neuronal cells that provide support and nutrition, maintain homeostasis, form myelin, and participate in signal transmission in the nervous system. In the human brain, glia cells are estimated to outnumber neurons by about 10 to 1.

Hybrid Training: :earning method that combines the supervised and unsupervised training of Connectionist Systems.

Synapse: Specialized junctions through which the cells of the nervous system signal to each other and to non-neuronal cells such as those in muscles or glands.

Association Rule Mining

Vasudha Bhatnagar
University of Delhi, India

Sarabjeet Kochhar
University of Delhi, India

INTRODUCTION

Data mining is a field encompassing study of the tools and techniques to assist humans in intelligently analyzing (mining) mountains of data. Data mining has found successful applications in many fields including sales and marketing, financial crime identification, portfolio management, medical diagnosis, manufacturing process management and health care improvement etc..

Data mining techniques can be classified as either descriptive or predictive techniques. Descriptive techniques summarize / characterize general properties of data, while predictive techniques construct a model from the historical data and use it to predict some characteristics of the future data. Association rule mining, sequence analysis and clustering are key descriptive data mining techniques, while classification and regression are predictive techniques.

The objective of this article is to introduce the problem of association rule mining and describe some approaches to solve the problem.

BACKGROUND

Association rule mining, one of the fundamental techniques of data mining, aims to extract interesting correlations, frequent patterns or causal structures among sets of items in data.

An association rule is of the form $X \rightarrow Y$ and indicates that the presence of items in the antecedent of rule (X) implies the presence of items in the consequent of rule (Y). For example, the rule *{PC, Color Printer}* $\rightarrow$ *{computer table}* implies that people who purchase a PC (personal computer) and a color printer also tend to purchase a computer table. These associations, however, are not based on the inherent characteristics of a domain (as in a functional dependency) but on the co-occur-

rence of data items in the dataset. Thus, association rule mining is a totally data driven technique.

Association rules have been successfully employed in numerous applications, some of which are listed below:

1. **Retail market analysis:** Discovery of association rules in retail data has been applied in departmental stores for floor planning, stock planning, focused marketing campaigns for product awareness, product promotion and customer retention.
2. **Web association analysis:** Association rules in web usage mining have been used to recommend related pages, discover web pages with common references, web pages with majority of same links (mirrors) and predictive caching. The knowledge is applied to improve web site design and speed up searches.
3. **Discovery of linked concepts:** Words or sentences that appear frequently together in documents are called linked concepts. Association rules can be used to discover linked concepts which further lead to the discovery of plagiarized text and the development of ontologies etc..

The problem of association rule mining (ARM) was introduced by Agrawal et al. (1993). Large databases of retail transactions called the market basket databases, which accumulate in departmental stores provided the motivation of ARM. The basket corresponds to a physical retail transaction in a departmental store and consists of the set of items a customer buys. These transactions are recorded in a database called the transaction database. The goal is to analyze the buying habits of customers by finding associations between the different items that customers place in their "shopping baskets". The discovered association rules can also be used by management to increase the effectiveness of

Figure 1. Boolean database and corresponding transaction database

TID	A	B	C	D	E
10	1	1	1	0	0
20	1	0	1	0	0
30	1	0	1	1	0
40	0	1	1	0	1
50	1	0	1	0	1

TID	Items
10	A, B,C
20	A, C
30	A, C, D
40	B, C, E
50	A, C, E

advertising, marketing, inventory management and reduce the associated costs.

The authors in (Agrawal et al., 1993) worked on a *boolean database* of transactions. Each record corresponds to a customer basket and contains transaction identifier (*TID*), transaction details and a *list of items* bought in the transaction. The list of items is represented by a boolean vector with a *one* denoting presence of corresponding item in the transaction and *zero* marking the absence. Figure 1 shows the *boolean database* of five transactions and the corresponding transaction database.

The problem of finding association rules is to find the columns with frequently co-occurring ones in the boolean database. However, most of the algorithms for ARM use the form of transaction database shown on the right. We give the mathematical formulation of the problem below.

MATHEMATICAL FORMULATION OF THE ARM PROBLEM

Let $I = \{i_1, i_2, ..., i_n\}$ denote a set of items and D designate a database of N transactions. A transaction $T \in D$ is a subset of I i.e. $T \subseteq I$ and is associated with a unique identifier *TID*.

An *itemset* is a collection of one or more items. X is an itemset if $X \subseteq I$. A transaction is said to contain an itemset X if $X \subseteq T$. A *k-itemset* is an itemset that contains k items.

An *association rule* is of the form $X \rightarrow Y$ *[Support, Confidence]* where $X \subset I$, $Y \subset I$, $X \cap Y = \varnothing$, and *Support* and *Confidence* are rule evaluation metrics.

Support of an itemset X is the fraction of transactions that contain X. It denotes the probability that a transaction contains X.

Support $(X) = \mathrm{P}(X) =$

$$\frac{\text{No. of transactions containing } X}{\text{Total number of transactions in } D}$$

Support of a rule $X \rightarrow Y$ in D is 's' if $s\%$ of transactions in D contain $X \cup Y$, and is computed as:

Support $(X \rightarrow Y) = \mathrm{P}(X \cup Y) =$

$$\frac{\text{No. of transactions containin } X \cup Y}{\text{Total number of transactions in } D}$$

Support indicates the extent of prevalence of a rule. A rule with low support value represents a rare event.

Confidence of a rule measures its strength and provides an indication of the reliability of prediction made by the rule. A rule $X \rightarrow Y$ has a confidence 'c' in D if $c\%$ of transactions in D that contain X also contain Y. It is computed as the conditional probability that Y occurs in a transaction, given X is present in the same transaction, i.e.

Confidence $(X \rightarrow Y) = \mathrm{P}(Y/X) =$

$$\frac{\mathrm{P}(X \cup Y)}{\mathrm{P}(X)}$$

Example 1: Consider the example database shown in Figure 2 (a). Here, I = {A, B, C, D, E}. Figures 2 (b) and 2 (c) show the computation of support and confidence for a rule.

Figure 2(a). Example of database transactions

TID	Items
10	A, B,C
20	A, C
30	A, C, D
40	B, C, E
50	A, C, E

Figure 2(b). Itemsets of size one and two

Figure 2(c). Computation of support and confidence

Sup(A): 4 (80%),
Supp(C): 5 (100%)
Sup (AB): 1 (20%)
Sup(AC): 4 (80%),
Sup (ABC): 1 (20%),
Sup (ABCD): 0 (0%)
Sup(ABCDE): 0 (0%)

Confidence (A→C) = 4/4 (100%)
Confidence (C→A) = 4/5 (80%)

The second criterion of interestingness is the strength of the rule. A rule which has confidence greater than the user specified minimum confidence threshold (*minconf*) is interesting to the user.

Confidence, however, can sometimes be misleading. For instance, the confidence of a rule can be high even if antecedent and consequent of the rule are independent. *Lift (*also called *Interest)* and *Conviction* of a rule are other commonly used measures for rule interestingness (Dunham, 2002). A suitable measure of rule strength needs to be identified for an application.

MINING OF ASSOCIATION RULES

Since an association rule is an implication among itemsets, the brute-force approach to association rule generation requires examining relationships between all possible item-sets. Such a process would typically involve counting the co-occurrence of all itemsets in D and subsequently generating rules from them.

For *n* items, there are 2^n possible itemsets that need to be counted. This may require not only prohibitive amount of memory, but also complex indexing of counters. Since the user is interested only in frequent rules, one needs to count only the frequent itemsets before generating association rules. Thus, the problem of mining association rules is decomposed into two phases:

Phase I: Discovery of frequent itemsets

Phase II: Generation of rules from the frequent itemsets discovered in phase I.

Various algorithms for ARM differ in their approaches to optimize the time and storage requirement for counting in the first phase. Despite reducing the search space by imposing *minsup* constraint, frequent itemset generation is still a computationally expensive process. The *Anti-monotone property* of support is an important tool to further reduce the search space and is used in most association rule algorithms (Agrawal et al., 1994). According to this property, support of an itemset never exceeds the support of any of its subsets i.e. if *X* is an itemset, then for each of its subsets *Y*, sup(X) <= sup(Y). This property makes the set of frequent itemsets *downward closed*. The task of rule

With *n* items in *I*, the total number of possible association rules is very large ($O(3^n)$). However, the majority of these rules (associations) existing in *D* are not interesting for the user. Interestingness measures are employed to reduce the number of rules discovered by the algorithms. Foremost criterion of interestingness is the high prevalence of both the item-sets and rules, which is specified by the user as minimum support value. An itemset (rule) whose support is greater than or equal to a user specified minimum support (*minsup)* threshold is called a *Frequent Itemset (rule).*

Figure 3. Generation of association rules from frequent itemsets (Adapted from Dunham, 2002)

```
Algorithm: Gen-Rules
Input: F – Set of frequent itemsets
        minconf – Minimum confidence threshold
Output: R – set of strong association rules
Process:
        R = ∅
        For each f ∈ F do
              For each x ⊂ f such that x ≠ ∅ do
                  If sup(f)/sup(x) >= minconf
                      R = R ∪{x → (f - x)}
```

generation is trivial once the set of frequent itemsets has been discovered.

Apriori algorithm uses a level-wise approach for discovering frequent itemsets (Agrawal et al., 1994). This algorithm makes multiple scans of the data. In each scan the algorithm generates and counts potential frequent itemsets (*candidates*). Candidate itemsets of size *k+1* are generated by joining the frequent itemsets of size *k*, and are pruned exploiting the anti-monotonic property. At the end of the scan the set of frequent itemsets is confirmed. The strategy incurs massive I/O costs and has motivated a number of variants which reduce the number of counters (*candidates*) and the scans (Brin et al., 1997; Lin et al., 1998). The FP-growth algorithm (Han et al., 2000) uses a pattern growth method to avoid the costly process of candidate generation and testing, and therefore is considered as a major milestone. The algorithm uses an in-memory, tree based data structure called FP-Tree to store the database in a compressed form. Two passes are made over the database to construct the prefix tree, which is then used to generate frequent patterns. Lattice based approaches have also been proposed for efficient discovery of frequent itemsets (Zaki et al., 1998).

The task of generating rules (Phase II) from a given frequent itemsets is rather straight forward (Figure 3). It boils down to enumerating all subsets of a frequent itemset and finding the ratio of support of each itemset w.r.t. the support of each of its subsets. The subsets, whose ratio is more than the *minconf*, qualify as strong rules and are reported to the user.

TYPES OF ASSOCIATION RULES

Popularity of ARM has led to its application on many types of data and application domains. Some specialized kinds of association rules have been reported in data mining literature (Han & Kamber 2006). We describe here some of the important types:

1. **Quantitative Association Rules:** Quantitative association rules introduced the notion of mining associations between numeric attributes in addition to the categorical ones (Srikant et al., 1996). The quantitative association rules can be derived by either mapping the attribute values to a set of consecutive integers or partitioning them into intervals. Example of a quantitative association rule:

Age (x, "30...39") ^ salary (x, "42...48K") → buys(x, "car") [1%, 75%]

2. **Multilevel Association Rules:** Many applications have an inherent *taxonomy (concept hierarchy)* among items (Figure 4). In such scenarios, association rules can be generated at different levels of the taxonomy, to capture knowledge at different levels of abstraction. As we move down the hierarchy (from generalized towards specialized values), the support of rules decreases and some rules may become uninteresting. However while climbing up the hierarchy some new rules may become interesting.

This gives rise to *multilevel association rules* (Han et al., 1999), which capture linkages between items or attributes at different levels of abstraction, i.e. at different levels of concept hierarchy. For example, the rule *{Brown Bread} → {Coke}* captures linkages between items at different levels of concept hierarchy. Multilevel rules can be mined either using the same support thresholds or different thresholds at different levels.

Selecting an appropriate level of abstraction can have a significant impact on the usefulness of the knowledge generated. Mining at a very high level of abstraction is likely to generate over-generalized rules, which may not be interesting, while mining at too low an abstraction level may lead to generation of highly specific rules.

A

Figure 4. Concept hierarchy for food
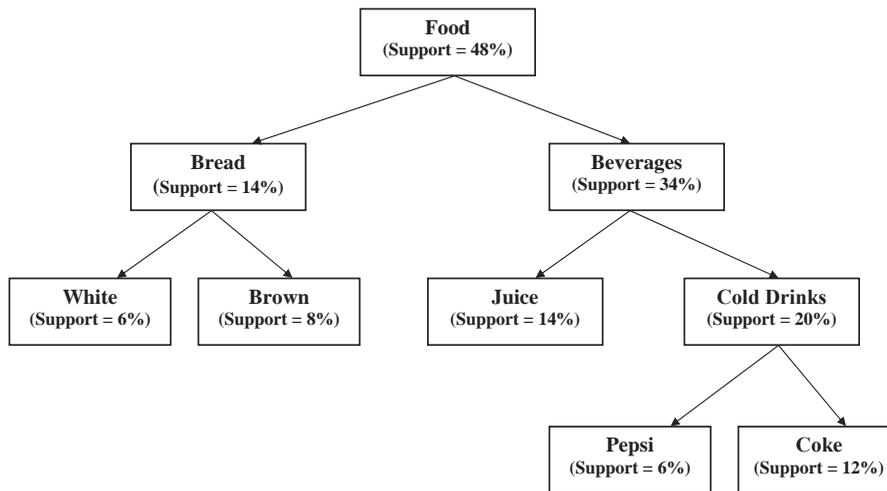
3. **Multidimensional Association Rules:** Association rules essentially expose intra-record linkages. If links exist between different values of the same attribute (dimension), the association is called single-dimensional association rule, while if linkages span multiple dimensions, the associations are called multi-dimensional association rules (Kamber et al., 1997). For example the following single dimensional rule contains conjuncts in both antecedent and consequent from a single attribute 'buys'.

Buys (X, "milk") and Buys(X, "butter") → Buys(X, "bread")

Multidimensional associations contain two or more predicates, for example

Age (X,"19-25") and Occupation(X, "student") → Buys(X, "coke")

4. **Association Rules in Streaming Data:** The aforementioned types of rules pertain to static data repositories. Mining rules from data streams, however, is far more complex (Hidber, 1999). The field is relatively recent and poses additional challenges such as one-look characteristics of data, limited main memory, continual updating of data and online processing so as to be able to make decisions on the fly (Cheng et al. 2007).

5. **Association Rule Mining with Multiple Minimum Supports**: In large departmental stores where the number of items is very large, using single support threshold sometimes does not yield interesting rules. Since buying trends for different items often vary vastly, the use of different support thresholds for different items is recommended (Liu et al., 1999).

6. **Negative Association Rules**: Typical association rules discover correlations between items that are bought during the transactions and are called positive association rules. Negative association rules discover implications in all the items, irrespective of whether they were bought or not. Negative association rules are useful in market-basket analysis to identify products that conflict with each other or complement each other. The importance of negative association rules was highlighted in (Brin et al., 1997; Wu et al. 2004).

FUTURE TRENDS

Though association analysis will continue to impact various scientific and business spheres through healthcare databases, financial databases, spatial databases, multimedia databases and time series databases etc., we investigate the role of association rule mining in some promising applications.

Business Intelligence (BI): BI converts raw data into personalized intelligence with the goal of increasing customer satisfaction, loyalty and product profitability. It integrates data from multiple sources across company, analyzes it and acts promptly on the results leading to competitive advantage and timely deployment of solutions.

Association analysis is one of the core tools that support BI. In the retail sector association analysis provides basis for point-of-sale data analysis, market basket analysis, resources and space management optimization. In the banking realm, BI exploits the study of associations to perform credit risk analysis, fraud detection, and customer retention. The credit companies benefit by fraud detection, monitoring buying patterns, scoring customer reliability and analyzing cross-sell.

Stream data mining: In contrast to the data in traditional static databases, a data stream is an ordered sequence of items that is continuous, unbounded, usually comes with high speed and has a data distribution that changes with time. As the number of applications on mining data streams grows rapidly, there is an increasing need to perform association rule mining on stream data.

Association rules are employed in the estimation of missing data in streams of data generated by sensors and frequency estimation of internet packet streams. Association rule mining is also useful for monitoring manufacturing flows to predict failure or generate reports based on web log streams.

Bioinformatics: Associations are employed in bioinformatics databases for identification of co-occurring gene sequences. They are also used to detect gene mutations. A gene is a segment of a DNA molecule that contains all the information required for the synthesis of a product. Any change in the DNA sequence of a gene (For example: Insertion, Deletion, Insertion/Deletion, Complex and Multiple Substitution) is termed gene mutation. The discovery of interesting association relationships among huge amount of gene mutations is important because it can help in determining the cause of mutation in tumours and diseases.

CONCLUSION

Association rule mining is an important data mining technique that was introduced to describe the intra record links in transactional databases. However, both academia and industry have seen the technology spell profitability and success due to its simplicity, ease of understanding and wide applicability. More than a decade later, the technology still promises to be the driving force for some new, challenging applications. No doubt that association rule mining has been regarded as one of the most significant contribution from the database community in KDD.

In this article, we introduced the notion of association rules, their applications and presented the mathematical formulation for the same. We also presented the major milestones in the development history of association rules. The general mining strategy, the related problems such as vastness of search space and the interestingness measures were also discussed. Finally, different types of association rules were described.

REFERENCES

Agrawal R., Imielinski T., & Swami A. (1993), Mining Association Rules Between Sets of Items in Large Databases, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207-216, Washington, D.C.

Agrawal R., & Srikant R. (1994), Fast Algorithms for Mining Association Rules in Large Databases, *Proceedings of the Twentieth International Conference on Very Large Databases*, pp. 487-499, Santiago, Chile

Hidber C. (1999), Online Association Rule Mining, in *proceedings of ACM SIGMOD International Conference on Management of Data, Philadephia, Pennsylvania*, pp.145-156

Han, J., Pei, J., & Yin, Y. (2000), Mining frequent patterns without candidate generation, *Proceedings of 2000 ACM SIGMOD Intl. Conference on Management of Data*, W. Chen, J. Naughton, & P. A. Bernstein, Eds. ACM Press, 1-12.

A

Savasere A., Omiecinski E., & Navathe S. (1995), An efficient algorithm for mining association rules in large databases. In *Proceedings of the 21st VLDB Conference*, pages 432-443, Zurich, Switzerland.

Srikant R. & Agrawal, R. (1996), Mining quantitative association rules in large relational tables, *Proceedings of the ACM SIGMOD Conference on Management of Data, Montreal*, Canada.

Srikant R. & Agrawal R (1995), Mining Generalized Association Rules, *Proceedings of. the 21st Int'l Conference on Very Large Databases*, Zurich, Switzerland.

Brin S., Motwani R. & Silverstein C (1997), Beyond market basket: Generalizing association rules to correlations, *Proceedings of SIGMOD*.

Dunham M.(2002), Data Mining: Introductory and Advanced Topics, Prentice Hall.

Brin S., Motwani R., Ullman J. D., & Tsur S. (1997), Dynamic itemset counting and implication rules for market basket data. SIGMOD Record (ACM Special Interest Group on Management of Data), 26(2):255.

Lin D. & Kedem Z. M. (1998), Pincer search: A new algorithm for discovering the maximum frequent sets, *Proceedings of the 6th Int'l Conference on Extending Database Technology (EDBT)*, Valencia, Spain.

Zaki M. J. & Ogihara M. (1998), Theoretical foundations of association rules, *Proceedings of 3rd SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'98)*, Seattle, Washington, USA.

Han J. & Kamber M. (2006), Data Mining: Concepts and Techniques, 2nd ed. Morgan Kaufmann Publishers.

Han J. & Fu Y. (1999), Mining Multiple-Level Association Rules in Large Databases. IEEE Trans. Knowl. Data Eng. 11(5): 798-804.

Kamber M., Han J. & Chiang J., (1997), Metarule-Guided Mining of Multi-Dimensional Association Rules Using Data Cubes. KDD 1997: 207-210.

Liu B., Hsu W., Ma Y., (1999) Mining Association Rules with Multiple Minimum Supports, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, San Diego, CA, USA.

Cheng J., Ke Y., Ng W. (2007), A Survey on Algorithms for Mining Frequent Itemsets Over Data Streams, in international journal of Knowledge and Information Systems.

Wu X., Zhang C., and Zhang S. (2004), Efficient mining of both positive and negative association rules. *ACM Trans. Inf. Syst.* 22, 3 (Jul. 2004), 381-405.

KEY TERMS

Association Rule: An implication of the form $X \rightarrow Y$, in a transactional data base with parameters support (s) and confidence (c). X and Y are set of items, s is the fraction of transactions containing $X \cup Y$ and c% of transactions containing X also contain Y.

Classification: Data mining technique that constructs a model (*classifier*) from historical data (training data) and uses it to predict the category of unseen tuples.

Clustering: Data mining technique to partition data objects into a set of groups such that the intra group similarity is maximized and inter group similarity is minimized.

Data Mining: Extraction of interesting, non-trivial, implicit, previously unknown and potentially useful information or patterns from data in large databases.

Data Stream: A continuous flow of data from a data source, e.g., a sensor, stock ticker, monitoring device, etc. A data stream is characterized by its unbounded size.

Descriptive Mining Technique: Data mining technique that induces a model describing the characteristics of data. These techniques are usually unsupervised and totally data driven.

Predictive Mining Technique: Data mining technique that induces a model from historical data in supervised manner, and uses the model to predict some characteristic of new data.

Automated Cryptanalysis

Otokar Grošek
Slovak University of Technology, Slovakia

Pavol Zajac
Slovak University of Technology, Slovakia

INTRODUCTION

Classical ciphers are used to encrypt plaintext messages written in a natural language in such a way that they are readable for sender or intended recipient only. Many classical ciphers can be broken by brute-force search through the key-space. One of the pertinent problems arising in automated cryptanalysis is the plaintext recognition. A computer should be able to decide which of many possible decrypts are meaningful. This can be accomplished by means of a text scoring function, based, e.g. on n-grams or other text statistics. A scoring function can also be used in conjunction with AI methods to speedup cryptanalysis.

BACKGROUND

Language recognition is a field of artificial intelligence studying how to employ computers to recognize language of a text. This is a simple task when we have enough amount of text with accents since they characterize used language with very high accuracy. Nowadays there are plenty of toolkits which automatically check/correct often both spelling and grammatical mistakes and errors. In connection with this we recall also the NIST Language Recognition Evaluation (LRE-05, LRE-07) as a part of an ongoing series of evaluations of language recognition technology. McMahon & Smith (1998) present an overview of natural language processing techniques based on statistical models.

We recall some basic notions from cryptography (see the article automated cryptanalysis of classical ciphers for more details). There is a reversible encryption rule (algorithm) how to transform plaintext to the ciphertext, and vice-versa. These algorithms depend on a secret parameter K called the key. The set of possible keys $\mathcal{K}$ is called the key-space. Input and output of these algorithms is a string of letters from plaintext, or ciphertext alphabet respectively. Both, sender as well as receiver, uses the same secret key, and the same encryption and decryption algorithms.

Cryptanalysis is a process of key recovery, or plaintext recovery without the knowledge of the key. In both cases we need a plaintext recognition subroutine which evaluates (with some probability) every candidate substring, whether it is a valid plaintext or not. Such automated text recognition requires an adequate model of a used language.

PLAINTEXT RECOGNITION FOR AUTOMATED CRYPTANALYSIS

In the process of automated cryptanalysis we decrypt the ciphertext with many possible keys to obtain candidate plaintexts. Most of the candidates are incorrect, having no meaning in a natural language. On the other hand, even the correct plaintext can be hard to recognize and with the wrong recognition routine can be missed altogether.

The basic type of algorithm suitable for automated cryptanalysis is a brute force attack. This attack is only feasible when key-space is searchable on computational resources available to an attacker. The average time needed to verify a candidate strongly influences the size of searchable key-space. Thus, the plaintext recognition is the most critical part of the algorithm from the performance point of view. On the other hand, only the most complex algorithms achieve really high accuracy of the plaintext recognition. Thus the complexity and accuracy of plaintext recognition algorithms must be carefully balanced.

A generic brute force algorithm with plaintext recognition can be described by the pseudo-code in Exhibit A.

Exhibit A.

INPUT:	ciphertext string $Y = y_0\,y_1\,y_2...y_n$
OUTPUT:	ordered sequence S of possible plaintexts with their scores

1. **Let** $S = \{\ \}$
2. **For each** key $K \in \mathcal{K}$ **do**
   2.1. Let $X = d_K(\ Y)$ be a candidate plaintext.
   2.2. Compute **negative test predicate** *filter(X)*. **If** predicate is **true**, continue with step 2.
   2.3. Compute **fast scoring function** *fastScore(X)*. **If** *fastScore(X) < LIMITF*, continue with step 2.
   2.4. Compute **precise scoring function** *score(X)*. **If** *score(X) < LIMIT*, continue with step 2.
   2.5. Let $S = S \cup \{<score(X), X> \}$
3. **Sort** S by key *score(X)* descending.
4. **Return** S.

Table 1. Performance of the three-layer decryption of a table-transposition cipher using a brute-force search. First filter was negative predicate-based, removing all decrypts with first 4 letters not forming a valid n-gram (about 90 % of texts were removed). Score was then computed as the count of valid tetragrams in the whole text. If this count was lower then given threshold (12), then the text was removed in the score-based filter. Finally, remaining texts were scored using the dictionary words.

Key-space Size	Negative filter	Score-based filter	Remaining texts	Total time [s]
9!	89.11%	10.82%	254	1.2
10!	89.15%	10.78%	2903	5.8
11!	88.08%	8.92%	239501	341
12!	90.10%	9.85%	1193512	746

Algorithm integrates the three layers of plaintext recognition, namely *negative test predicate, fast scoring function* and *precise scoring function,* as a three-layer filter. The final scoring function is also used to sort the outputs. First filter should be very fast, with very low error probability. Fast score should be easy to compute, but it is not required to precisely identify the correct plaintext. Correct plaintext recognition is the role of precise scoring function. In the algorithm, the best score is the highest one. If the score is computed in the opposite meaning, the algorithm must be rewritten accordingly.

In some cases, we can integrate a fast scoring function within the negative test or with the precise scoring, leading to two-layer filters, as in (Zajac, 2006a). It is also possible to use even more steps of predicate-based and score-based filtering, respectively. However, experiments show that the proposed architecture of three-layers is the most flexible, and more layers can even lead to performance decrease. Experimental results are shown in Table 1.

Negative Filtering

The goal of the *negative test predicate* is to identify candidate texts that are NOT plaintext (with very high probability, ideally with certainty). People can clearly recognize the wrong text just by looking at it. It is in the area of artificial intelligence to implement this ability in computers. However, most nowadays AI methods (e.g. neural networks) seem to be too slow, to be applicable in this stage of a brute-force algorithm, as every text must be evaluated with this predicate.

Most of the methods for fast negative text filtering are based on prohibited *n*-grams. As an *n*-gram

we would only consider a sequence of *n* consecutive letters. If the alphabet size is *N*, then it is possible to create N^n possible *n*-grams. For higher *n,* only a small fraction of them can appear in valid text in a given language (Zajac, 2006b). By using a lexical tree or lookup table, it is easy (and fast) to verify, whether a given *n*-gram is valid or not. Thus a natural test is to check every *n*-gram in the text, whether it is valid or not. There are two basic problems arising with this approach – the real plaintext can contain (intentionally) misspelled, uncommon or foreign words, and thus our *n*-gram database can be incomplete. We can limit our test to some specific patterns, e.g. too long run of consecutive vowels/consonants. These patterns can be checked in time dependent on the plaintext candidate length. A filter can also be based on checking only a few *n*-grams on a fixed or random position in the text, e.g. the first four letters.

The rule for rejecting texts should be based on the exact type of the cipher we are trying to decipher. For example, if the first four letters of the decrypted plaintext does not depend on some part of the key, the filter based only on their validity would not be effective. An interesting question is, whether it is possible to create a system which can effectively learn its filter rules from existing decrypted texts, even in the process of decryption.

Scoring Functions

With the negative filter step we can eliminate around 90% of candidate texts or more. The number of texts to be verified is still very huge, and thus we need to apply more precise methods of plaintext recognition. We use a scoring function that assigns a quantity – score – to every text that has survived elimination in previous steps. Here the higher score means higher likeness that a given text is a valid plaintext. For each scoring function we can assign a threshold, and it should be very improbable that a valid plaintext have score under this threshold. Actual threshold value can either be found experimentally (by evaluating large number of real texts), or can be based on a statistical analysis. Speed of the scoring function can be determined by using classical algorithm complexity estimates. Precision of the scoring can be defined by means of separation of valid and invalid plaintexts, respectively. There is a trade-off involved in scoring, as faster scoring functions are less precise and vice-versa. Thus we apply

two scoring functions: one that is fast but less precise, with lower threshold value, and one that is very precise, but harder to compute.

An example of scoring function distributions can be found in Figures 1 and 2. Scoring function in Figure 1 is much more precise than in Figure 2, but computational time required for evaluation is doubled. Moreover, scoring function in Figure 1 was created from a reduced dictionary fitted to a given ciphertext. Evaluation based on a complete dictionary is slower, more difficult to implement, and can even be less precise.

Scoring functions can be based on dictionary words, *n*-grams statistics, or other specific statistics. It is difficult to provide a one-fits-all scoring function, as decryption process for different cipher types has impact on actual scoring function results. E.g. when trying to decrypt a transposition cipher, we already know which letters appear with which frequency, and thus letter frequency based statistics do not play any role in scoring. On the other hand they are quite significant for substitution ciphers. Most common universal scoring functions are (see also Ganesan & Sherman, 1993):

1. **Number of dictionary words in the text / fraction of meaningful text**
    Scoring based on dictionary words is very precise, if we have a large enough dictionary. Even if not every word in the hidden message is in our dictionary, it is very improbable that some incorrect decryption contains some larger fraction of a text composed of dictionary words. Removing short words from dictionary can increase the precision. Another possibility is to use weights based on word length as in (Russell, Clark & Stepney, 2003). Dictionary words can be found using lexical trees. In some languages, we should use dictionary of word stems, instead of the whole words. Speed of evaluation depends on the length of the text and the average length of words in dictionary, respectively.

2. **Rank distance of *n*-grams**
    Rank of the *n*-gram is its position depending on order based *n*-gram frequencies (merged for all *n* up to given bound dependent on language). This method is used (Cavnar & Trenkle, 1994) in fast automated language recognition: compute ranks of *n*-grams of given text and compare it with ranks of significant *n*-grams obtained from large corpus of different languages. Correct language should

Figure 1. Distribution of score among 9! possible decrypts (table transposition cipher with key given by a permutation of 9 columns), ciphertext size is 90 characters. Score was computed as a weighted sum of lengths of (reduced) dictionary words found in the text. Single highest score belongs to the correct plaintext.
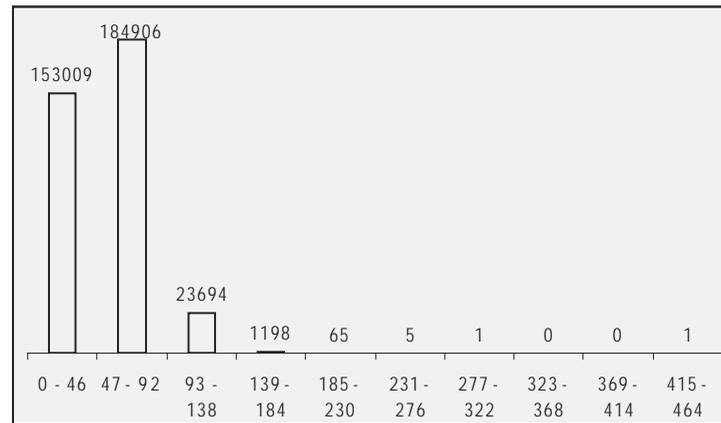
have the smallest distance. Even if this method can be adapted for plaintext recognition, e.g. by creating "random" corpus, or encrypted corpus, it does not seem practical.

3. **Frequency distance of *n*-gram statistics**

Score of the text can be estimated from the difference of measured frequency of *n*-grams and estimated frequencies from large corpus (Clark, 1998; Spillman, Janssen, Nelson & Kepner, 1993). We suppose that correct plaintext would have the smallest distance from corpus statistics. However due to statistical properties of the text, this is not always true for short or specific texts. Thus the precision of this scoring function is higher for longer texts. Speed of the evaluation depends on the text size and size of *n*.

4. **Scoring tables for *n*-grams**

If we consider the statistics of all *n*-grams, we will see that most *n*-grams contribute a very small value to the final score. We can consider contribution of only the most common *n*-grams. For a given language (and a given ciphertext) we can prepare a table of *n*-gram scores with fixed number of entries. Score is evaluated as the sum of scores assigned to *n*-grams in a given candidate text (Clark & Dawson, 1998). We can assign both positive scores for common valid *n*-grams, and negative scores for common invalid/supposedly rare *n*-grams. However, precision of this method

is very low especially for transposition ciphers. In our experiments the scores have normal distribution, and usually the correct plaintext does not have highest possible values. On the other hand, this scoring function is easy to evaluate, and can be customized for a given ciphertext. Thus it can be used as a fast scoring function *fastScore(X)*.

5. **Index of coincidence (and other similar statistics)**

Index of coincidence (Friedman 1920), denoted by I_C, is a suitable and fast statistics applicable for ciphers that modify the letter frequency. The notion comes from probability of the same letter occurring in two different texts at the same position. Encrypted texts are considered random, and have (normalized) index of coincidence near to the 1.0. On the other hand, a plaintext has much higher I_C near the value expected for a given language and alphabet. For English language the expected value is 1.73. As with all statistics based on scoring function, its precision is influenced by the length of the text. Index of coincidence is most suitable for polyalphabetic ciphers, where encryption depends on the position of the letter in the text (e.g. Vigenère cipher). It can be adapted to other cipher types (Bagnall, McKeown & Rayward-Smith, 1997), e.g. for transposition ciphers when considering that alphabet is created by all possible *n*-grams with some $n > 1$.

A

Figure 2. Distribution of score among 9! possible decrypts (table transposition cipher with key given by a permutation of 9 columns), ciphertext size is 90 characters. Score is the count of digrams with frequency higher than 1% (in Slovak language) in a given text. The correct plaintext has scored 73.
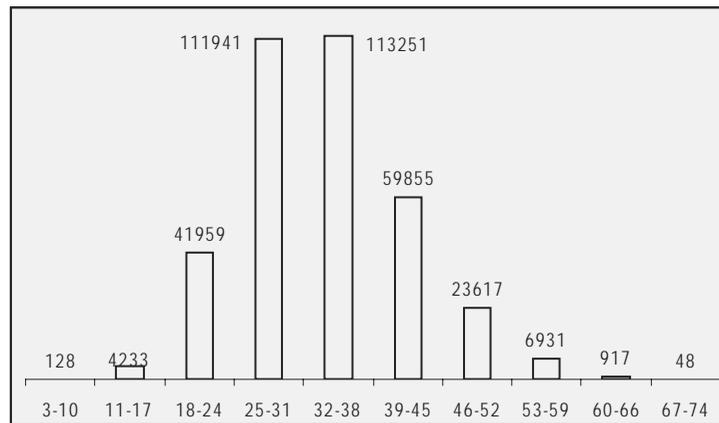
FUTURE TRENDS

Language processing and recognition have applications in various areas outside cryptanalysis (OCR, automatic translation...). Some cryptanalytic techniques can be generalized for these fields. E.g. some letters or groups of letters are often replaced by another in scanned documents. Thus correcting these documents is similar to cryptanalysis of randomized substitution ciphers. With Artificial Intelligence research new insights can be gained into a structure of natural language that can help further in cryptanalysis. Cryptanalysis is also strongly related to automatic translation efforts.

Some open problems that need to be addressed by language recognition suitable for cryptanalysis of classical ciphers are the following:

- How the text recognition should be integrated with decryption process to give feedback, e.g. on partially decrypted words, to estimate a new key, etc. This is especially true, if we use more advanced search heuristic than brute-force search through the key-space. This can also be viewed as a generalization of results of Peleg & Rosenfeld (1979).
- How the syntax and semantics of the language can help in text recognition and key search, respectively.

- How various encodings and writing systems influence cryptanalysis. Specific issues arise when dealing with different writing systems (Atkinson 1985; August 1989 and 1990).
- How to correctly recognize text with intentional misspellings and special code words.

Another set of problems arises when different natural languages are used, like the language recognition, specific alphabets, impact of diacritical marks, etc. Our research shows that the language of the message encrypted by substitution cipher can be recognized even without decryption (Zajac, 2006b). It is even possible to use dictionary of a different (although similar) language in decryption process. It is an interesting research question whether it is possible to create completely general language recognition function (or restricted to some family of languages) usable for cryptanalysis.

Plaintext recognition in cryptanalysis can be also seen as a specific information retrieval problem (Manning, Raghavan & Schütze 2008). Multilanguage information retrieval is targeting similar problems to the problems presented above (see e.g. McNamee, 2006). The research in these areas can clearly influence each other in the future.

CONCLUSION

This article summarizes the usage and restrictions for language processing in the context of cryptanalysis of classical ciphers. Their application usually differs according to a character of the analyzed cipher systems, although we have presented some common techniques that can be easily adapted for a specific situation. Most cryptanalytic attacks require very fast language recognition, but on the other hand, great speed often causes inaccurate results, up to the point of unrecognizable decrypts. The role of the Artificial Intelligence research is to find faster and more precise language predicates and combine them to a useful plaintext recognition system.

REFERENCES

Atkinson, R. (1985). *Ciphers in Oriental Languages.* Cryptologia, 9(4), 373-380.

August, D. A. (1989). *Cryptography and Exploitation of Chinese Manual Cryptosystems - Part I: The Encoding Problem.* Cryptologia, 13(4), 289-302.

August, D. A. (1990). *Cryptography and Exploitation of Chinese Manual Cryptosystems - Part II: The Encrypting Problem.* Cryptologia, 14(1), 61-78.

Bagnall,T. & McKeown, G. P. & Rayward-Smith, V. J. (1997). *The cryptanalysis of a three rotor machine using a genetic algorithm.* In Thomas Back, editor, Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97), San Francisco, CA. Morgan Kaufmann.

Cavnar, W.B., & Trenkle, J.M. (1994). *N-gram-based text categorization.* Proceedings of the Third Symposium on Document Analysis and Info, 161-175.

Clark, A. J. (1998). *Optimisation Heuristics for Cryptology.* PhD thesis, Information Security Research Center, Faculty of Information Technology, Queensland University of Technology.

Clark, A. & Dawson, E. (1998). *Optimisation Heuristics for the Automated Cryptanalysis of Classical Ciphers.* Journal of Combinatorial Mathematics and Combinatorial Computing, vol. 28, 63-86.

Friedman, W. F. (1920). *The Index of Coincidence and Its Applications in Cryptography*, Riverbank Publication No. 22, Riverbank Labs., Geneva, Ill..

Ganesan, R. & Sherman, A. (1993). *Statistical techniques for language recognition: An introduction and guide for cryptanalysts*, Cryptologia,17(4), 321-366.

Manning, C.D. & Raghavan P. & Schütze, H. (2008) Introduction to Information Retrieval, Cambridge University Press.

McMahon, J. & Smith, F.J. (1998). *A Review of Statistical Language Processing Techniques.* Artificial Intelligence Review 12 (5), 347-391.

McNamee, P. (2006). *Why You Should Use N-grams for Multilingual Information Retrieval*. UMBC eBiquity Talk. http://www.umiacs.umd.edu/research/CLIP/colloq/abstracts/2006-10-18-slides.pdf

Peleg, S. & Rosenfeld, A. (1979). *Breaking Substitution Ciphers Using a Relaxation Algorithm.* Communications of the ACM 22(11), 598--605.

Russell, M. D. & Clark, J. A. & Stepney, S. (2003). *Making the most of two heuristics: Breaking transposition ciphers with ants*. Proceedings of IEEE Congress on Evolutionary Computation (CEC 2003). IEEE Press, 2653--2658.

Spillman, R. & Janssen, M. & Nelson, B. & Kepner, M. (1993). *Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers*. Cryptologia, 17(1), pp. 31-44.

Zajac, P. (2006a). *Automated Attacks on Transposition Ciphers*. Begabtenförderung im MINT Bereich 14, 61-76.

Zajac, P. (2006b). *Ciphertext language identification*. Journal of Electrical Engineering, 57 (7/s), 26--29.

KEY TERMS

Brute-Force Attack: Exhaustive cryptanalytic technique that searches the whole key-space to find the correct key.

Candidate Text: The text that was obtained by application of decryption algorithm on ciphertext using some key $k \in \mathcal{K}$. If k is the correct key (or the equiva-

lent key to) K, then candidate text is a valid plaintext x, otherwise it is a text encrypted by concatenation of $d_k(e_K(x))$.

Ciphertext: The encrypted text, a string of letters from alphabet $\mathcal{C}$ of a given cryptosystem by a given key $K \in \mathcal{K}$.

Classical Cipher: A classical cipher system is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where $\mathcal{P}, \mathcal{C}$, define plaintext and ciphertext alphabet, $\mathcal{K}$ is the set of possible keys, and for each $K \in \mathcal{K}$, there exists an encryption algorithm $e_K \in \mathcal{E}$, and a corresponding decryption algorithm $d_K \in \mathcal{D}$ such that $d_K(e_K(x)) = x$ for every input $x \in \mathcal{P}$ and $K \in \mathcal{K}$.

Cryptanalysis: Is a process of trying to decrypt given ciphertext and/or find the key without, or with only partial knowledge of the key. It is also a research area studying techniques of cryptanalysis.

Key-Space: Set of all possible keys for a given ciphertext. Key-space can be limited to a subspace of the whole $\mathcal{K}$ by some prior knowledge.

Plaintext: The unencrypted text, a string of letters from alphabet $\mathcal{P}$ of a given cryptosystem.

Plaintexts Filter: An algorithm, or predicate, used to determine, which texts are not valid plaintexts. Ideal plaintexts filter never produces answer INVALID for a correct plaintext.

Scoring Function: Scoring function is used to evaluate fitness of a candidate text for a key $k \in \mathcal{K}$. Ideal scoring function has global extreme in the correct plaintext, i.e. when $k = K$.

Automated Cryptanalysis of Classical Ciphers

Otokar Grošek
Slovak University of Technology, Slovakia

Pavol Zajac
Slovak University of Technology, Slovakia

INTRODUCTION

Classical ciphers are used to encrypt plaintext messages written in a natural language in such a way that they are readable for sender or intended recipient only. Many classical ciphers can be broken by brute-force search through the key-space. Methods of artificial intelligence, such as optimization heuristics, can be used to narrow the search space, to speed-up text processing and text recognition in the cryptanalytic process. Here we present a broad overview of different AI techniques usable in cryptanalysis of classical ciphers. Specific methods to effectively recognize the correctly decrypted text among many possible decrypts are discussed in the next part Automated cryptanalysis – Language processing.

BACKGROUND

Cryptanalysis can be seen as an effort to translate a ciphertext (an encrypted text) to a human language. Cryptanalysis can thus be related to the computational linguistics. This area originated with efforts in the United States in the 1950s to have computers automatically translate texts from foreign languages into English, particularly Russian scientific journals. Nowadays it is a field of study devoted to developing algorithms and software for intelligently processing language data. Systematic (public) efforts to automate cryptanalysis using computers can be traced to first papers written in late '70s (see e.g. Schatz, 1977). However, the research area has still many open problems, closely connected to an area of Artificial Intelligence. It can be concluded from the current state-of-the-art, that although computers are very useful in many cryptanalytic tasks, a human intelligence is still essential in complete cryptanalysis.

For convenience of a reader we recall some basic notions from cryptography. Very thorough survey of classical ciphers is written by Kahn (1974). A message to be encrypted (plaintext) is written in the lowercase alphabet $\mathcal{P} = \{a, b, c \dots x, y, z\}$. The encrypted message (ciphertext) is written in uppercase alphabet $C = \{A, B, C \dots X, Y, Z\}$. Different alphabets are used in order to better distinguish plaintext and ciphertext, respectively. In fact these alphabets are the same.

There is a reversible encryption rule (algorithm) how to transform the plaintext to the ciphertext, and vice-versa. These algorithms depend on a secret parameter K called the key. The set of possible keys $\mathcal{K}$ is called the key-space. Input and output of these algorithms is a string of letters from respective alphabets, $\mathcal{P}^*$ and C^*. Both, sender as well as receiver, uses the same secret key, and the same encryption and decryption algorithms.

There are three basic classical systems to encrypt a message, namely a substitution, a transposition, and a running key. In a substitution cipher a string of letters is replaced by another string of letters using prescribed substitution of single letters, e.g. left 'a' to 'A', replacing letter 'b' by letter 'N', letter 'c' by letter 'G', etc. A transposition cipher rearranges order of letters according to a secret key K. Unlike substitution ciphers the frequency of letters in the plaintext and ciphertext remains the same. This characteristic is used in recognizing that the text was encrypted by some transposition cipher. A typical running key cipher is to derive from a main key K the running key $K_0 K_1 K_2 \dots K_n$. If $\mathcal{P} = C = \mathcal{K}$ is a group, then simply $y_i = e_K(x_i) = x_i + K_i$.

Thus it is convenient to define a ciphering algorithm for classical ciphers as follows:

Definition 1: A classical cipher system is a five-tuple $(\mathcal{P}, C, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where the following conditions are satisfied:

1. $\mathcal{P}$ is a finite set of a plaintext alphabet, and $\mathcal{P}^*$ the set of all finite strings of symbols from $\mathcal{P}$.
2. C is a finite set of a ciphertext alphabet, and C^* the set of all finite strings of symbols from C..
3. $\mathcal{K}$ is a finite set of possible keys.
4. For each $K \in \mathcal{K}$, there is an encryption algorithm $e_K \in \mathcal{E}$, and a corresponding decryption algorithm $d_K \in \mathcal{D}$ such that $d_K(e_K(x)) = x$ for every input $x \in \mathcal{P}$ and $K \in \mathcal{K}$.
5. The ciphering algorithm assigns to any finite string $x_0 x_1 x_2 ... x_n$ from $\mathcal{P}^*$ the resulting ciphertext string $y_0 y_1 y_2 ... y_n$ from C^*, where $y_i = e_K(x_i)$. The actual key may, or need not depend on the index i.

Another typical case for $\mathcal{P}$, and C, are r-tuples of the Latin alphabet. For transposition ciphers, the key is periodically repeated for r-tuples. For substitution ciphers of r-tuples, the key is an r-tuple of keys. In the case of running keys, there is another key stream generator $g: \mathcal{K} \times \mathcal{P} \rightarrow \mathcal{K}$ which generates from the initial key K, and possibly from the plaintext $x_0 x_1 x_2 ... x_{n-1}$ the actual key K_n.

For classical ciphers, there are two typical situations when we try to recover the plaintext:

1. Let the input to decryption algorithm $d_K \in \mathcal{D}$ with unknown key K be a ciphertext string $y_0 y_1 y_2 ... y_n$ from C^*, where $y_i = e_K(x_i)$. Our aim is to find the plaintext string $x_0 x_1 x_2 ... x_n$ from $\mathcal{P}^*$. Thus in each execution an algorithm is searching through Key-space $\mathcal{K}$.
2. The decryption algorithm $d_K \in \mathcal{D}$ and key K are unknown. Our aim is to find for the ciphertext string $y_0 y_1 y_2 ... y_n$ from C^*, where $y_i = e_K(x_i)$, the plaintext string $x_0 x_1 x_2 ... x_n$ from $\mathcal{P}^*$. This requires a different algorithm than the actual $d_K \in \mathcal{D}$, as well as some additional information. Usually there is available another ciphertext, say $z_0 z_1 z_2 ... z_n$ from C^*. Thus in each execution an algorithm is searching through possible substitutions which are suitable for both ciphertexts.

In both cases we need a plaintext recognition subroutine which evaluates a candidate substring of length v for a possible plaintext, say $c_t c_{1+t} c_{2+t} ... c_{v+t} := x_t x_{1+t} x_{2+t} ... x_{v+t}$. Such automated text recognition needs an adequate model of a used language.

AUTOMATED CRYPTANALYSIS

There are two straightforward methods for automated cryptanalysis. Unfortunately none of them is for longer strings applicable in practice. The first one is for transposition ciphers. When no other information about the cipher is known, we can use a general method, called anagramming, to decipher the message. In this method we are trying to assemble the meaningful string (anagram) from the ciphertext. This is accomplished by arranging the letters to words from the dictionary. When we find the meaningful word we process the rest of the message in the same way. When we are not able to create more meaningful words, we retrace our steps, and try other possible words until the whole meaningful anagram is found.

The second, and very similar, is for the substitution ciphers. Here we are trying to assemble the meaningful string (anagram) from the ciphertext by searching through all possible substitutions of letters to get words from dictionary of the used language. Although the size of the key-space is large, automated cryptanalysis uses many other methods based, e.g. on frequency distribution of letters. Automated cryptanalysis of simple substitution ciphers can decrypt most of the messages both with known word boundaries (Carrol & Martin, 1986), and without this information (Ramesh, Athithan & Thiruvengadam, 1993; Jakobsen, 1995). There are other classical ciphers, where transposition or substitution depends not only on the actual key, but also on a position within a block of letters of the string.

For effective automated cryptanalysis at least two layers of plaintext candidate processing, filtering and scoring, are required. Better results are achieved by additional filtering layers. This of course increases computational complexity. Bellow we give an overview of these filtering layers.

Automated Brute Force Attacks

The basic type of algorithm suitable for automated cryptanalysis is a brute force attack. As we have to search the whole key-space, this attack is only feasible when key-space is "not too large". Exact quantification of the searchable key-space depends on computational resources available to an attacker, and the average time needed to verify a candidate for decrypted text. Thus, the plaintext recognition is the most critical part of the algorithm from the performance point of view.

A

On the other hand, only the most complex algorithms achieve really high accuracy of plaintext recognition. Thus the careful balance of the complexity of plaintext recognition algorithms and its accuracy is required. It is unlikely that automated cryptanalysis produces only one possible result, but it is possible to limit the set of possible decrypts to a manageable size. Reported results should be sorted according to their probability of being the true plaintext.

A generic brute force algorithm with plaintext recognition can be described by the pseudo-code in Exhibit A.

We have identified three layers of plaintext recognition, namely *negative test predicate, fast scoring function* and *precise scoring function.* All three functions are used as a three-layer filter, and final scoring function is also used to sort the outputs. First filter should be very fast, and should have very low error probability. Fast score should be easy to compute, but it is not required to precisely identify the correct plaintext. Correct plaintext recognition is the role of precise scoring function. In the algorithm, the best score is the highest one. If the score is computed in the opposite meaning, the algorithm must be rewritten accordingly.

In some cases, we can integrate a fast scoring function within negative test or with the precise scoring, leading to two-layer filters, as in (Zajac, 2006a). It is also possible to use even more steps of predicate-based and score-based filtering, respectively. However, experiments show that the proposed architecture of three-layers is the most flexible, and more layers can even lead to performance decrease. Scoring and fil-

tering is described in-depth in the article Automated cryptanalysis – Language processing.

Applications of Artificial Intelligence Methods

Artificial Intelligence (AI) methods can be used in four main areas of the automated cryptanalysis:

1. *Plaintext recognition:* The goal of the AI is to supply negative predicates that filter out wrong decrypts, and scoring functions that assess the text's likeness to natural language.
2. *Key-search heuristics:* The goal of the AI is to provide heuristics to speed-up the decryption process either by constraining the key-space, or by guiding the selection of next keys to be tried in the decryption. This area is most often researched, as it can provide clear experimental results, and meaningful evaluation.
3. *Plaintext estimation:* The goal of the AI is to estimate the meaning of the plaintext from the partial decryption, or to estimate some parts of the plaintext based on external data (e.g. a sender of a ciphertext, historical and geographic context, specific grammatical rules etc.) Estimated parts of the plaintext can then lead to much easier complete decryption. This area of research is mainly unexplored, and plaintext estimation is done by the cryptanalyst.
4. *Automatic security evaluation:* The goal of the cryptanalysis is not only to break ciphers and to

Exhibit A.

INPUT: ciphertext string $Y = y_0\ y_1\ y_2...y_n$
OUTPUT: ordered sequence S of possible plaintexts with their scores
1. Let $S = \{ \}$
2. For each key $K \in \mathcal{K}$ **do**
2.1. Let $X = d_K(\ Y)$ be a candidate plaintext.
2.2. Compute **negative test predicate** *filter(X).* **If** predicate is **true**, continue with step 2.
2.3. Compute **fast scoring function** *fastScore(X).* **If** *fastScore(X) < LIMITF*, continue with step 2.
2.4. Compute **precise scoring function** *score(X).* **If** *score(X) < LIMIT*, continue with step 2.
2.5. Let $S = S \cup \{<score(X), X>\}$
3. Sort S by key *score(X)* descending.
4. Return S.

learn secrets, but it is also used when creating new ciphers to evaluate their security. Although most classical ciphers are already "outdated", their cryptanalysis is still important, e.g. in teaching the modern computer security principles. When teaching classical ciphers, it is useful to have an AI tool (e.g. an expert system), that can automate the evaluation of cipher security (at least under some weaker assumptions). Although much work is done in automatic evaluation of modern security protocols, we are unaware of some tools to evaluate "classical" cipher designs.

Area that is best researched is the area of *Key-search heuristics.* It immediately follows from the fact that brute force search through the whole key-space can be considered as a very crude method of decryption. Most classical ciphers were not designed with careful consideration of the text statistics. We can assign score for each key in the key-space that is correlated with the probability that text decrypted by given key is the plaintext. The score, when considered over the key-space, certainly have some local maxima, which can lead either immediately to a meaningful plaintext, or a text from which plaintext is easily guessed. Thus it can be useful to consider various relaxation techniques to search through the key-space with the goal of maximizing scoring function. One of the earliest demonstrations of relaxation techniques for breaking substitution ciphers are presented by Peleg & Rosenfeld (1979) and Hunter & McKenzie (1983). Successful attacks applicable for many classical ciphers can be implemented using basic hill climbing, through tabu search, simulated annealing and applications of genetic/evolution algorithms (Clark & Dawson, 1998). Genetic algorithms have achieved many successes in breaking classical ciphers as demonstrated by Mathews (1993), or Clark (1994), and can even break a rotor machine (Bagnall, McKeown & Rayward-Smith, 1997). Russell, Clark & Stepney (1998) present anagramming attack using a solver based on an ant colony optimisation algorithm.

These types of attack try to converge to the correct key by small changes of the actual key. Success rate of the attacks is usually measured by the fraction of the reconstructed key and/or text. Relaxation methods can find with a high probability the keys, or the plaintext approximations, even if it is not feasible to search the whole key-space. The success mainly depends on the ciphertext size, since the scoring is usually statistics-based. One of the unexplored challenges is to consider application of multiple relaxation techniques. First heuristic can be used to shrink the key-space, and then either the brute-force search or another heuristic is used with more precision to finish the decryption.

FUTURE TRENDS

The results obtained strongly depend on the size of the ciphertext, and decryptions are usually only partial. Techniques of the automated cryptanalysis also need to be fitted to a given problem. E.g. attacks on substitution ciphers can use individual letter statistics, but for attacks intended for transposition ciphers these statistics are invariant and make no sense in using. Automated cryptanalysis is usually studied only in context of these two main types of ciphers, but there is a broad area of unexplored problems concerning different classical cipher types, such as running key type ciphers. Specific uses of AI techniques can fail for some cryptosystems as pointed by Wagner, S., Affenzeller, M. & Schragl, D. (2004). Cryptanalysis also depends on the language (Zajac, 2006b), although there are some notable exceptions when considering similar languages.

As the computational power increases, even just recently used ciphers, like Data Encryption Standard (DES), are becoming subject of automated cryptanalysis (e.g. Nalini & Raghavendra Rao, 2007). Beside application of heuristics to cryptanalysis, a lot of further research is required in areas of plaintext estimation and automatic security evaluation. An expert system that would cover these areas and connect them with AI for plaintext recognition and search heuristics can be a strong tool to teach computer security or to help forensic analysis or historical studies involving encrypted materials.

CONCLUSION

This article is concerned with an automated cryptanalysis of classical ciphers, where classical ciphers are considered as a cipher from before WW2, or pencil-and-paper ciphers. Optimization heuristics are quite successful in attacks targeted to these ciphers, but they usually cannot be made fully-automatic. Their application usually differs according to a character of the analysed cipher systems. An important research

direction is extending the techniques from classical cryptanalysis to automated decryption of modern digital cryptosystems. Another important problem is to create set of fully-automatic cryptanalytic tools or a complete expert system that can be adapted to various types of ciphers and languages.

REFERENCES

Bagnall,T. & McKeown, G. P. & Rayward-Smith, V. J. (1997). *The cryptanalysis of a three rotor machine using a genetic algorithm*. In Thomas Back, editor, Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97), San Francisco, CA. Morgan Kaufmann.

Carrol, J. & Martin, S. (1986). *The automated cryptanalysis of substitution ciphers*. Cryptologia, 10(4). 193-209.

Clark, A. (1994). *Modern optimisation algorithms for cryptanalysis*. In Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems, November 29 - December 2, 258-262.

Clark, A. & Dawson, E. (1998). *Optimisation Heuristics for the Automated Cryptanalysis of Classical Ciphers*. Journal of Combinatorial Mathematics and Combinatorial Computing, vol. 28, 63-86.

Hunter, D.G.N. & McKenzie, A. R. (1983). *Experiments with Relaxation Algorithms for Breaking Simple Substitution Ciphers*. Comput. J. 26(1), 68-71

Jakobsen, T. (1995). *A fast method for cryptanalysis of substitution ciphers*. Cryptologia, 19(3). pp. 265-274.

Kahn, D. (1974): *The codebreakers*. Wiedenfeld and Nicolson, London.

Matthews, R.A.J. (1993). *The use of genetic algorithms in cryptanalysis.* Cryptologia, 17(4), 187-201.

Nalini, N. & Raghavendra Rao, A. (2007). *Attacks of simple block ciphers via efficient heuristics.* Information Sciences: an International Journal 177 (12), 2553--2569.

Peleg, S. & Rosenfeld, A. (1979). *Breaking Substitution Ciphers Using a Relaxation Algorithm.* Communications of the ACM 22(11), 598--605.

Ramesh, R.S. & Athithan, G. & Thiruvengadam, K. (1993). *An automated approach to solve simple substitution ciphers.* Cryptologia, 17(2), 202-218.

Russell, M. D. & Clark, J. A. & Stepney, S. (2003). *Making the most of two heuristics: Breaking transposition ciphers with ants*. Proceedings of IEEE Congress on Evolutionary Computation (CEC 2003). IEEE Press, 2653--2658.

Schatz, B. (1977). *Automated analysis of cryptograms*. Cryptologia, 1(2), 265-274. Also in: Cryptology: yesterday, today, and tomorrow, Artech House 1987, ISBN: 0-89006-253-6.

Wagner, S. & Affenzeller, M. & Schragl, D. (2004). *Traps and Dangers when Modelling Problems for Genetic Algorithms*. Cybernetics and Systems, pp. 79-84.

Zajac, P. (2006a). *Automated Attacks on Transposition Ciphers*. Begabtenförderung im MINT Bereich 14, 61-76.

Zajac, P. (2006b). *Ciphertext language identification***. Journal of Electrical Engineering, 57 (7/s), 26--29.**

KEY TERMS

Brute-Force Attack: Exhaustive cryptanalytic technique that searches the whole key-space to find the correct key.

Ciphertext: The encrypted text, a string of letters from alphabet C of a given cryptosystem by a given key $K \in \mathcal{K}$.

Classical Cipher: A classical cipher system is a five-tuple $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$, where $\mathcal{P}, \mathcal{C},$ define plaintext and ciphertext alphabet, $\mathcal{K}$ is the set of possible keys, and for each $K \in \mathcal{K}$ there exists an encryption algorithm $e_K \in \mathcal{E},$ and a corresponding decryption algorithm $d_K \in \mathcal{D}$ such that $d_K(e_K(x)) = x$ for every input $x \in \mathcal{P}$ and $K \in \mathcal{K}$.

Cryptanalysis: Is a process of trying to decrypt given ciphertext and/or find the key without, or with only partial knowledge of the key. It is also a research area studying techniques of cryptanalysis.

Key-Space: Set of all possible keys for a given ciphertext. Key-space can be limited to a subspace of the whole $\mathcal{K}$ by some prior knowledge.

Plaintext: The unencrypted text, a string of letters from alphabet $\mathcal{P}$ of a given cryptosystem.

Relaxation Attack: Cryptanalytic technique that searches the key-space by incremental updates of the candidate key(s). It usually applies the knowledge of previous trial decryption(s) to change some parts of the key.

Automatic Classification of Impact-Echo Spectra I

Addisson Salazar
iTEAM, Polytechnic University of Valencia, Spain

Arturo Serrano
iTEAM, Polytechnic University of Valencia, Spain

INTRODUCTION

We investigate the application of artificial neural networks (ANNs) to the classification of spectra from impact-echo signals. In this paper we provide analyses from simulated signals and the second part paper details results of lab experiments.

The data set for this research consists of sonic and ultrasonic impact-echo signal spectra obtained from 100 3D-finite element models. These spectra, along with a categorization of the materials among homogeneous and defective classes depending on the kind of material defects, were used to develop supervised neural network classifiers. Four levels of complexity were proposed for classification of materials as: material condition, kind of defect, defect orientation and defect dimension. Results from Multilayer Perceptron (MLP) and Radial Basis Function (RBF) neural networks with Linear Discriminant Analysis (LDA), and k-Nearest Neighbours (kNN) algorithms (Duda, Hart, & Stork, 2000), (Bishop C.M., 2004) are compared. Suitable results for LDA and RBF were obtained.

The impact-echo is a technique for non-destructive evaluation based on monitoring the surface motion resulting from a short-duration mechanical impact. It has been widely used in applications of concrete structures in civil engineering. Cross-sectional resonant modes in impact-echo signals have been analyzed in elements of different shapes, such as, circular and square beams, beams with empty ducts or cement fillings, etc. In addition, frequency analyses of the displacement of the fundamental frequency to lower values for detection of cracks have been studied (Sansalone & Street, 1997), (Carino, 2001).

The impact-echo wave propagation can be analyzed from transient and stationary behaviour. The excitation signal (the impact) produces a short transient stage where the first P (normal stress), S (shear stress) and Rayleigh (superficial) waves arrive to the sensors; afterward the wave propagation phenomenon becomes stationary and a manifold of different mixtures of waves including various changes of S-wave to P-wave propagation mode and viceversa arrive to the sensors. Patterns of waveform displacements in this latter stage are known as the resonant modes of the material. The spectra of impact-echo signals provide of information for classification based on resonant modes the inspected materials. The classification tree approached in this paper has four levels from global to detailed classes with up to 12 classes in the lowest level. The levels are: (i) <u>Material condition</u>: homogeneous, one defect, multiple defects, (ii) <u>Kind of defect</u>: homogeneous, hole, crack, multiple defects, (iii) <u>Defect orientation</u>: homogeneous, hole in axis X or axis Y, crack in planes XY, ZY, or XZ, multiple defects, and (iv) <u>Defect dimension</u>: homogeneous, passing through and half passing through types of holes and cracks of level iii, multiple defects. Some examples of defective models are in Figure 1.

BACKGROUND

Neural networks applications in impact-echo testing include: detect flaws on concrete slabs, combining spectra of numerical simulations and real signals for network training (Pratt & Sansalone, 1992), identification of unilaterally working sublayer cracks using numerically generated waveforms as network inputs (Stavroulakis, 1999), classification of concrete slabs in solid and defective (containing void or delamination), use of training features extracted from many repetitions of impact-echo experiments on three specimens to be classified in three classes (Xiang & Tso, 2002), and to predict shallow crack depths in asphalt pavements using features from an extensive real signal

dataset (Mei, 2004). All these studies used multilayer perceptron neural network and monosensor impact-echo systems.

In a recent work, we classified impact-echo data by neural networks using temporal and frequency features extracted from the signals, finding that the better features were frequency features (Salazar, Unió, Serrano, & Gosalbez, 2007). Thus the present work is focused in exploiting only spectra information of the impact-echo signals. These spectra contain a large amount of redundant information. We applied Principal Component Analysis (PCA) to spectra for compressing and removing noise. The proposed classification problem and the use of spectra PCA components as classification features are a new proposal in application of neural networks to impact-echo testing.

There is evidence that the first components of PCA retain essentially all of the useful information and this compression optimally removes noise and can be used to identify unusual spectra (Bailer-Jones, 1996), (Bailer-Jones, Irwin, & Hippel, 1998), (Xu et al., 2004). The principal components represent sources of variance in the data. The projection of the pth spectrum onto the kth principal component is known as the admixture coefficient $a_{k,p}$. The most significant principal components contain those features which are most strongly correlated in many of the spectra. It follows that noise (which is uncorrelated with any other features by definition) will be represented in the less significant components. Thus by retaining only the more significant components to represent the spectra we achieve a data compression that preferentially remove noise. The reduced reconstruction, $\mathbf{y}_p$ of the pth spectrum $\mathbf{x}_p$, is obtained by using only the first r principal components to reconstruct the spectrum, i.e.

$$\mathbf{y}_p = \overline{\mathbf{x}} + \sum_{k=1}^{k=r} a_{k,p}\mathbf{u}_k, \qquad r < N, \qquad (1)$$

where $\overline{\mathbf{x}}$ is the mean spectrum which is subtracted from the spectra before the eigenvectors are calculated, and $\mathbf{u}_k$ is the kth principal component. $\overline{\mathbf{x}}$ can be considered as the zeroth eigenvector, although the degree of variance it explains depends on the specific data set and may be much less than that explained by the first eigenvectors.

Let ε_p be the error incurred in using this reduced reconstruction. By definition $\mathbf{x}_p = \mathbf{y}_p + \varepsilon_p$, so

$$\varepsilon_p = \sum_{k=r+1}^{k=N} a_{k,p}\mathbf{u}_k \ . \qquad (2)$$

RECOGNITION OF DEFECT PATTERNS IN IMPACT-ECHO SPECTRA -SIMULATIONS

Impact-Echo Signals

Simulated signals came from full transient dynamic analysis of 100 3D finite element models of simulated parallelepiped-shape material of 0.07x0.05x0.22m. (width, height and length) supported to one third and two thirds of the block length (direction *z*). Figure 1 shows different examples of the models of defective pieces. From the transient analysis the dynamic response of the material structure (time-varying displacements in the structure) under the action of a transient load is estimated. The transient load, i.e. the hammer impact, was simulated by applying a force-time history of a half sine wave with a period of 64μs as a uniform pressure load on two elements at the centre of the model front face. The elastic material constants for the simulated material were: density 2700 kg/m3, elasticity modulus 69500 Mpa. and Poisson's ratio 0.22.

Elements having dimensions of about 0.01 m. were used in the models. These elements can accurately capture the frequency response up to 40 kHz. Surface displacement waveforms were taken from the simulation results at 7 nodes in different locations on the material surface, see Figure 1a. Signals consisted of 5000 samples recorded at a sampling frequency of 100 kHz. To make possible to compare simulations with experiments, the second derivative of the displacement was calculated to work with accelerations, since the sensors available for experiments were mono-axial accelerometers. These accelerations were measured in the normal direction to the plane of the material surface accordingly to the configuration of the sensors in Figure 1a.

Feature Extraction and Selection

We investigate if the changes in the spectra, particularly in the zones of the fundamental frequencies, are related

Figure 1. Finite element models with different defects and 7-sensor configuration
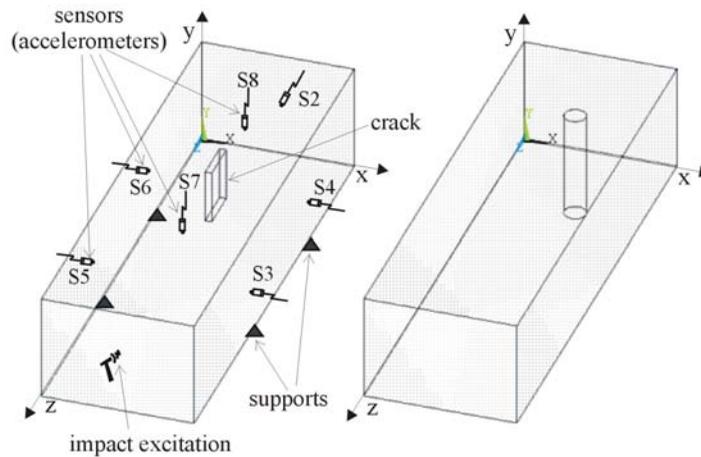
1a. Half-passing through crack oriented in plane ZY　　　　*1b. passing through hole oriented in axis Y*

with the shape, orientation and dimension of the defects. The information of the spectra for each channel consists of $n/2$ values as half of the number of points used to calculate the Fast Fourier Transform (FFT). Due to the 7-channel impact-echo system setup applied, the number of data available for each impact-echo test was $7*n/2$, e.g. for a FFT calculated with 256 points, 896 values would be available as entries for classifiers. This high number of entries could be unsuitable for the training stage of neural networks. Considering impact-echo signal spectra redundancy, PCA was applied in two steps. At first step, PCA was applied to the spectra of each channel as a feature extraction method. At second step, PCA was applied to the component set (spectra compressed) obtained in the first step for all the channels and records as dimensionality reduction and feature selection method. Thus, a compressed and representative pattern of the spectra for the multichannel impact-echo inspection was obtained.

The size of the FFT employed was 1024 points since using less points the resolution was not good enough for classifications. Once the spectra were estimated for all the models they were grouped and normalized by maximum per channel. There were considered three options to establish the number of components at the first PCA step: select a number of components that explain a minimum of the variance in the data, or a number of components such the variance increment is minimum, or a fixed number of components. The

first two options could estimate a variable number of components per channel, and they could select more components for the channels with 'worst' signals, i.e. signals with low signal to noise relation (SNR), due to problems in measuring (e.g. bad contact in the interface sensor and material). Thus we select a fixed number of components=20 per channel, that explained more than 95% of the data variance for each of the channels, so the total number of components was $7*20=140$ for one model.

The initial entries for the classification stage were then 140 features (spectra components) for the 100 simulation models. For simulations 20 replicates for each model were added that corresponded to the repetitions performed in the experiments. The replicates were generated using random Gaussian noise with 0.1-standard deviation of the original signals; then total of records for simulations was 2000 with 140 spectra components.

PCA was applied again to reduce the dimensionality of the classification space and to select the best spectra features for classification. After some preliminary tests, 50 was set as a number of components for classification. Using this number of components, the explained variance was 98%. With the 50 sorted components obtained, an iterative process of classification varying the number of components was applied using LDA and kNN as classifiers. The curve described by the set of classification error and number of components (5,10,15,…,50)

194

A

values has an inflection point where the information provided for the components perform the best classification. Following this feature selection process, a reduced set of features ('better' spectra components) was obtained. Those features were used as entries for ANNs, improving the performance of the classification, instead of using all the spectra components. The number of selected components for ANN classification varied from 20 to 30, depending on classification level (material condition, kind of defect, defect orientation, defect dimension).

The classification proceeded applying the Leave-One-Out method, avoiding records of replicas or repetitions of a test piece were in the training stage of that piece, so generalization of pattern learning was forced. Thus some of the records used in training and test corresponded to models or specimens with the same kind of defect but located in different positions, and the rest of records corresponded to others kind of defective pieces. Results presented in next sections are referring to mean error in testing stage.

Simulation Results

Figure 2a shows the results of classification by kNN and LDA with linear, Mahalanobis, and quadratic distances for simulations at level 4 of the classification tree. The best percentage of classification success (75.9) is obtained by LDA-quadratic and LDA-Mahalanobis with 25 components. Those components were selected and used as inputs for the input layer of the networks. One hidden layer was used (different number of neurons were tried to obtain the best configuration of the neuron number at this layer), and the number of neurons at the output layer was set as the number of classes, depending on the classification level. A validation stage and resilient propagation training method were used in classifications with MLP. The spread parameter was tuned for RBF, Figure 2b shows how the spread affects the classification results in the "defect dimension level", and in this case the minimum error (0.31) is for spread value 1.6.

Summarised general results by different classification methods for simulations are showed in Table 1. The best classification performance is obtained by LDA with quadratic distance, but results of RBF are fairly comparable. Due to classes are not equally-probable at each level, general results are weighted by class probability, see Figure 3. Homogeneous class was

completely distinguishable and multiple-defects class was the worst classified at every classification levels. The percentage of success could be very much higher by increasing classification success for multiple-defect class. This fact was caused because the multiple-defects models consisted in models with various cracks, and it yield confusion between the crack and multiple-defect classes. The percentage of success decreases for more complex classifications, with RBF lowest performance of 69% for 12 classes.

FUTURE TRENDS

The proposed methodology was tested with particular kind of material and defects and configuration of multichannel testing. It could be tested using models and specimens of different materials, sizes, sensor configurations, and signal processing parameters.

There exist several techniques and algorithms of classification that can be explored for the proposed problem. Recently a model of independent component analysis (ICA) was proposed for impact-echo (Salazar, Vergara, Igual, Gosalbez, & Miralles, 2004), and new classifiers based on mixtures of ICAs have been proposed (Salazar, Vergara, Igual, & Gosalbez, 2005), (Salazar, Vergara, Igual, & Serrano, 2007), that include issues as semisupervision in training stage. The use of prior knowledge in the training stage is critical in order to obtain suitable models for different kind of classifications. Those kind of techniques could give more understating on how labelled and labelled data change model learned by the classifier. In addition more research is needed on the shape of the classification space (impact-echo signal spectra), outlier probability, and decision region of the classes for the proposed problem.

CONCLUSION

We demonstrate the feasibility of using neural networks to extract patterns of different kinds of defects from impact-echo signal spectra in simulations. The methodology used was very restricted because there was only one piece for a defect in certain localization in the bulk and it was not in the training stage, so classifier had to assign the right class with the patterns of pieces of the same class in other localizations. Results could

Figure 2. LDA, kNN results and tuning of RBF parameter at Simulations, level 4 of classification
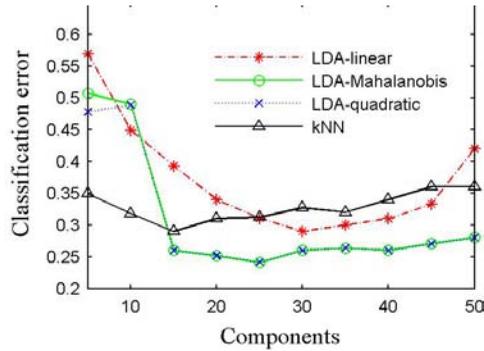
2a. LDA, kNN results for simulations at fourth level of classifi-
cation
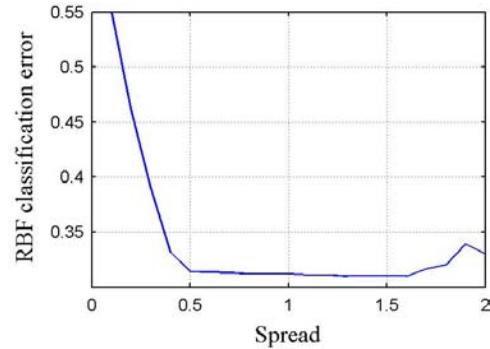
2b. RBF spread tuning for simulations at fourth level of clas-
sification

Table 1. Summarised classification results for simulations

	Error (%)	Level 1 (3 classes)	Level 2 (4 classes)	Level 3 (7 classes)	Level 4 (12 classes)
Simulations	LDA-L	6	13	30	29
	LDA-Q	8	9	19	24.1
	LDA-M	11.6	9	19	24.1
	kNN	8	14	25	29
	MLP	9	16	31	39
	RBF	8	17	26	31

be used to implement the proposed method in real applications of quality evaluation of materials; in those applications the database collected during reasonable time could have samples similar to the tested piece, making easier the classification process.

REFERENCES

Bailer-Jones, C. (1996). *Neural Network Classifiation of Stellar Spectra.* University of Cambridge.

Bailer-Jones, C., Irwin, M., & Hippel, T. (1998). Automated classification of stellar spectra - II. Two-dimensional classification with neural networks and principal components analysis. *Monthly Notices of the Royal Astronomical Society, 298,* 361-377.

Bishop C.M. (2004). *Neural newtworks for pattern recognition.* Oxford: Oxford University Press.

Carino, N. J. (2001). The impact-echo method: an overview. In Structures Congress and Exposition (Ed.), (pp. 1-18).

Duda, R., Hart, P. E., & Stork, D. G. (2000). *Pattern classification.* (2 ed.) New York: Wiley-Interscience .

Mei, X. (2004). Neural network for rapid depth evaluation of shallow cracks in asphalt pavements. *Computer-aided civil and infrastructure engineering, 19,* 223-230.

Pratt, D. & Sansalone, M. (1992). Impact-echo signal interpretation using artificial intelligence. *ACI Materials Journal, 89,* 178-187.

A

Figure 3. Percentages of success in classifications by RBF

Salazar, A., Unió, J., Serrano, A., & Gosalbez, J. (2007). Neural Networks for Defect Detection in Non-Destructive Evaluation by Sonic Signals. *Lecture Notes in Computer Science, 4507,* 638-645.

Salazar, A., Vergara, L., Igual, J., & Gosalbez, J. (2005). Blind source separation for classification and detection of flaws in impact-echo testing. *Mechanical Systems and Signal Processing, 19,* 1312-1325.

Salazar, A., Vergara, L., Igual, J., Gosalbez, J., & Miralles, R. (2004). ICA Model Applied to Multichannel Non-destructive Evaluation by Impact-echo. *Lecture Notes in Computer Science, 3195,* 470-477.

Salazar, A., Vergara, L., Igual, J., & Serrano, A. (2007). Learning Hierarchies from ICA Mixtures. In I. 2. 20th International Joint Conference on Neural Networks (Ed.).

Sansalone, M. & Street, W. (1997). *Impact-echo: Non-destructive evaluation of concrete and masonry.* New York: Bullbrier Press.

Stavroulakis, G. E. (1999). Impact-echo from a unilateral interlayer crack. LCP-BEM modelling and neural identification. *Engineering Fracture Mechanics, 62,* 165-184.

Xiang, Y. & Tso, S. K. (2002). Detection and classification of flaws in concrete structure using bispectra and neural networks. *NDT&E International, 35,* 19-27.

Xu, R., Nguyen, H., Sobol, P., Wang, S. L., Wu, A., & Johnson, K. E. (2004). Application of Principal Component Analysis to the FTIR Spectra of Disk Lubricant to Study Lube–Carbon Interactions. *IEEE Transactions on Magnetics, 40,* 3186-3189.

KEY TERMS

Artificial Neural Network (ANN): A mathematical model inspired in biological neural networks. The units are called neurons connected in various input, hidden and output layers. For a specific stimulus (numerical data at the input layer) some neurons are activated following an activation function and producing numerical output. Thus ANN is trained, storing the learned model in weight matrices of the neurons. This kind of processing has demonstrated to be suitable to find nonlinear relationships in data, being more flexible in some applications than models extracted by linear decomposition techniques.

Finite Element Method (FEM): It is a numerical analysis technique to obtain solutions to the differential equations that describe, or approximately describe a wide variety of problems. The underlying premise of FEM states that a complicated domain can be sub-divided into a series of smaller regions (the finite elements) in which the differential equations are approximately

197

Automatic Classification of Impact-Echo Spectra I

solved. By assembling the set of equations for each region, the behavior over the entire problem domain is determined.

Impact-Echo Testing: A non-destructive evaluation procedure based on monitoring the surface motion resulting from a short-duration mechanical impact. From analyses of the vibrations measured by sensors, a diagnosis of the material condition can be obtained.

Non-Destructive Evaluation (NDE): NDE, ND Testing or ND Inspection techniques are used in quality control of materials. Those techniques do not destroy the test object and extract information on the internal structure of the object. To detect different defects such as cracking and corrosion, there are different methods of testing available, such as X-ray (where cracks show up on the film), ultrasound (where cracks show up as an echo blip on the screen) and impact-echo (cracks are detected by changes in the resonance modes of the object).

Pattern Recognition: An important area of research concerned to discover or identify automatically figures, characters, shapes, forms, and patterns without active human participation in the decision process. It is also related with classify data in categories. Classification consists in learning a model for separating the data categories, that kind of *machine learning* can be approached using statistical (parametric or no-parametric models) or heuristic techniques. If some prior information is given in learning process, it is called supervised or semi-supervised, else it is called unsupervised.

Principal Component Analysis (PCA): A method for achieving a dimensionality reduction. It represents a set of *N*-dimensional data by means of their projections onto a set of *r* optimally defined axes (principal components). As these axes form an orthogonal set, PCA yields a data linear transformation. Principal components represent sources of variance in the data. Thus the most significant principal components show those data features which vary the most.

Signal Spectra: Set of frequency components decomposed from an original signal in time domain. There exist several techniques to map a function in time domain to frequency domain as Fourier and Wavelet transforms, and its inverse transforms that allow reconstructing the original signal.

Automatic Classification of Impact–Echo Spectra II

Addisson Salazar
iTEAM, Polytechnic University of Valencia, Spain

Arturo Serrano
iTEAM, Polytechnic University of Valencia, Spain

INTRODUCTION

We study the application of artificial neural networks (ANNs) to the classification of spectra from impact-echo signals. In this paper we focus on analyses from experiments. Simulation results are covered in paper I.

Impact-echo is a procedure from Non-Destructive Evaluation where a material is excited by a hammer impact which produces a response from the material microstructure. This response is sensed by a set of transducers located on material surface. Measured signals contain backscattering from grain microstructure and information of flaws in the material inspected (Sansalone & Street, 1997). The physical phenomenon of impact-echo corresponds to wave propagation in solids. When a disturbance (stress or displacement) is applied suddenly at a point on the surface of a solid, such as by impact, the disturbance propagates through the solid as three different types of stress waves: a P-wave, an S-wave, and an R-wave. The P-wave is associated with the propagation of normal stress and the S-wave is associated with shear stress, both of them propagate into the solid along spherical wave fronts. In addition, a surface wave, or Rayleigh wave (R-wave) travels throughout a circular wave front along the material surface (Carino, 2001).

After a transient period where the first waves arrive, wave propagation becomes stationary in resonant modes of the material that vary depending on the defects inside the material. In defective materials propagated waves have to surround the defects and their energy decreases, and multiple reflections and diffraction with the defect borders become reflected waves (Sansalone, Carino, & Hsu, 1998). Depending on the observation time and the sampling frequency used in the experiments we may be interested in analyzing the transient or the stationary stage of the wave propagation in im-

pact-echo tests. Usually with high resolution in time, analyzes of wave propagation velocity can give useful information, for instance, to build a tomography of a material inspected from different locations. Considering the sampling frequency that we used in the experiments (100 kHz), a feature extracted from the signal as the wave propagation velocity is not accurate enough to discern between homogeneous and different kind of defective materials.

The data set for this research consists of sonic and ultrasonic impact-echo signal (1-27 kHz) spectra obtained from 84 parallelepiped-shape (7x5x22cm. width, height and length) lab specimens of aluminium alloy series 2000. These spectra, along with a categorization of the quality of materials among homogeneous, one-defect and multiple-defect classes were used to develop supervised neural network classifiers. We show that neural networks yield good classifications (<15% error) of the materials in four levels of classification detail as material condition, kind of defect, defect orientation and defect dimension. Results for Multilayer Perceptron (MLP) and Radial Basis Function (RBF) neural networks, Linear Discriminant Analysis (LDA), and k-Nearest Neighbours (kNN) algorithms (Duda, Hart, & Stork, 2000), (Bishop C.M., 2004) are presented. Figure 1 shows the scheme of categories proposed as a hierarchical layout with different levels of knowledge on the material defects (the percentage of success in classification is explained in Experimental Result section).

Figure 1. Classification tree with percentages of success in classification by RBF network. Numbers in brackets are results for simulations (paper I). General results are weighted by class probability since classes are not equally-probable.
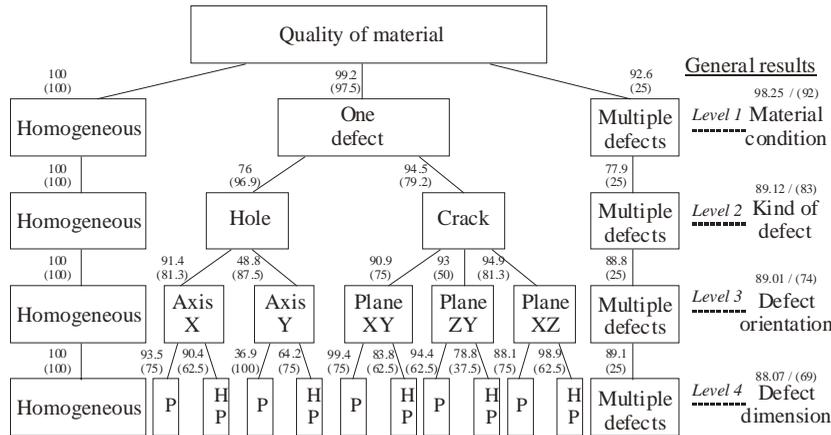
BACKGROUND

The phenomenon of volumetric wave propagation in impact-eco can be modelled by means of the following two equations (Cheeke J.D., 2002),

$$\frac{\partial T_{ij}}{\partial x_j} = \rho_0 \frac{\partial^2 u_i}{\partial t^2} \qquad (1)$$

$$T_{ij} = c_{ijkl} S_{kl} \qquad (2)$$

where

ρ_0: Material density.

u_i: Length elongation with respect to starting point in force direction.

$\frac{\partial T_{ij}}{\partial x_j}$: Force variation in i direction due to deformations in j directions.

c_{ijkl}: Elastic constant tensor (Hooke's law).

S_{kl}: Strain or relative volume change under deformation in face l in direction k in unitary cube that represents a material element.

Thus force variation in the direction i due to face stresses in j directions of the material elementary cube, is equal to the mass per volume (density) times the strain acceleration (Newton's third law in tensorial form). To derive an analytical solution to problems that involve stress wave propagation in delimited solids is very complicated, so bibliography on this subject is not very extensive. Numeric models such as the Finite Element Method (FEM) can be used to obtain an approximation to the material theoretical response (Abraham O, Leonard C., Cote P., & Piwakowski B., 2000).

There are several studies that used the impact-echo signals in frequency domain to detect the existence of defects in materials (Sansalone et al., 1997), (Hill, McHung, & Turner, 2000), (Sansalone, Lin, & Street, 1998). It has been demonstrated that a sequence of tones and harmonics appears in the spectra, they are fundamental modes of propagation that travel inside the material (block-shape material) and its frequencies depend on the shape and size of the material inspected by impact-echo. According to the block face where a sensor is located, some or others fundamental modes are captured. However, other tones are formed by the reflections of the waves with the defects in the material, and their frequencies are related with the deepness of the flaws. In addition, the presence of defects causes shifting of the fundamental mode frequencies due to diffractions.

MLP neural network has been applied to impact-echo in mono-sensor configurations (using only one accelerometer) to detect flaws on concrete slabs (Pratt & Sansalone, 1992), identification of unilaterally working sublayer cracks (Stavroulakis, 1999), classification of concrete slabs in solid and defective (Xiang & Tso, 2002). Those applications used a few number of ex-

periments and many repetitions or combined simulated with experimental signals, so its results may be verified because of probable overfitting. Other application is to predict shallow crack depths in asphalt pavements using features from an extensive real signal dataset (Mei, 2004). Recently, we provided an application of MLP, RBF and LVQ to classification tree proposed here using temporal and frequency features extracted from the signals, finding that the better features were frequency features (Salazar, Unió, Serrano, & Gosalbez, 2007).

In this paper we demonstrate the suitability of PCA application on the impact-echo signal spectra to obtain complex classifications in real experiments. The first components of PCA retain essentially all of the useful information and this compression optimally removes noise. The principal components represent sources of variance in the data. Thus the most significant spectra principal components show those features which vary the most between the spectra: it is important to realise that the principal components do not simply represent strong features. The principal components are eigenvectors of a symmetric matrix; they are simple rotations in the *N*-dimensional data space of the original axes on which the spectra are defined, thus they resemble the spectra (Bailer-Jones, 1996), (Bailer-Jones, Irwin, & Hippel, 1998), (Xu et al., 2004).

RECOGNITION OF DEFECT PATTERNS IN IMPACT-ECHO SPECTRA-EXPERIMENTS

Impact-Echo Signals

The equipment used in experiments was composed of: an instrumented hammer 084A14 PCB, 7 mono-axial accelerometers 353B17 PCB, a data acquisition module NI 6067E, a ICP signal conditioner F482A18 and a notebook for signal processing and control. The sample frequency in signal acquisition was 100,000 kHz, and observation time recorded was 30 ms. Figure 2a shows a photograph of the equipment employed in experiments, note that a 7x5x22cm. specimen with sensors positioned is being tested. Figure 2b shows a layout of the sensor locations on the surface of the piece (1 sensor at the back face, 4 sensors at the side faces, and 2 sensors at the top face), supports, and place of the impact. Sensors S4, S6, S8 are located at one third and S3, S5, S7 are located at two thirds of the piece length trough axis *Z*. S2 are in the middle of the opposite face to the impact.

The defects consisted in holes in the form of 10 mm. Ø cylinders, and cracks in the form of 5 mm. parallelepipeds with different orientations through the axes (*X, Y*) and planes (*XY, ZY, XZ*) of the material block. The dimensions of the defects were two: passing and half-passing through. Figure 2b shows a diagram of a defect of the class "half-passing trough crack oriented in plane *ZY*". The complete set of defective materials analyzed is depicted in Figure 1.

Figure 2. Experimental setup and sensor configuration
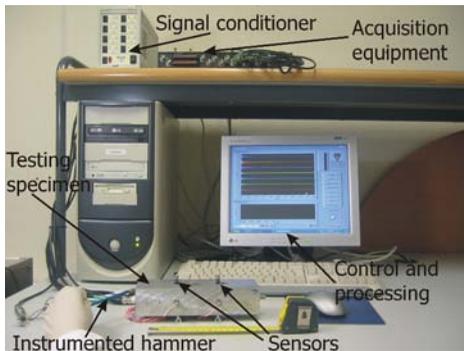
2a. Equipment
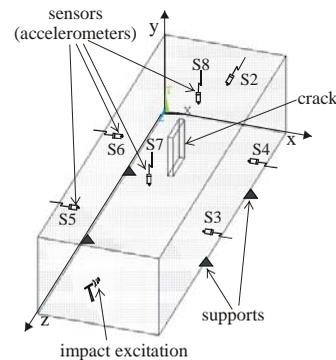
2b. Sensor configuration layout

Feature Extraction and Selection

The methodology followed for feature extraction, feature selection, reduction of dimensionality and classification in the impact-echo signal spectra was the applied in Paper I. After signal acquisition a four-stage procedure was followed: feature extraction, dimensionality reduction, feature selection, and classification with ANNs.

In the feature extraction stage, a 1024-points FFT was applied to the measured signals and these spectra were compressed by PCA, selecting the first 20 components per each channel. Thus entries for the dimensionality reduction stage were 140 components (7channelsx20) for the 84 lab specimens. For each experiment (specimen) were performed around 22 repetitions, so the total of records was 1881 for experiments each one with 140 spectra components. In the dimensionality reduction stage PCA reduced the 140 spectra components to 50 spectra components with a 92% explained variance. This matrix of 50 selected components by 1881 records was the input for a feature selection process which objective was to found the "best" number of components for classification. Then various tests of classification using LDA and kNN varying the number of components from 5 to 50 by increments of 5 were applied. The components corresponding to the best percentage of success in classification with kNN and LDA were selected as entries for the stage of classification with MLP and RBF. The number of spectra components varied from 10 to 30 depending on the classification level. Parameters as spread for RBF, and the number of neurons in the hidden layer for MLP were tuned to obtain the best classification percentage of success of the ANNs.

All the classification used Leave-One-Out method. Repetitions of a piece in testing were not used in its training stage to avoid classifier to memorize the pieces instead of to generalize patterns. Table 1 shows summarised results for all the classifiers applied at the different levels of classification, these results refer to mean error in testing stage.

Experimental Results

General results of classifications for experiments in Table 1 show the RBF as the best classifier, improving its performance near to 20% with regard to simulation results in paper I at the more complex level of classification (12 classes). The percentage of classification success improved for every class at each level, particularly for multiple-defect class from 25% up to 92.6% at first level and 89.1% at fourth level, see Figure 1. In experiments, specimens with multiple-defects were prepared combining cracks and holes, so there was not much confusion with multiple-defect and one-defect classes.

Real experiments of impact-echo involved random variables in its execution, as the force injected in the impact excitation, and the position of the sensors that can vary from piece to piece due to they are manually controlled. Those variables yield repetitions of the experiments with its corresponding signal spectra that separate better class regions than Gaussian noise used to obtain replicates of the simulated model signals.

The results of experiment classifications confirm the feasibility of using neural networks for pattern recognition of defects in impact-echo signals. Table 2 contains the confusion matrix at level "defect orientation". Homogeneous class is perfectly classified, and all the rest

Table 1. Summarised classification results for experiments

Error (%)	Level 1 (3 classes)	Level 2 (4 classes)	Level 3 (7 classes)	Level 4 (12 classes)
LDA-L	7.8	18.3	28.5	39.6
LDA-Q	6.7	15.3	20.8	26.3
LDA-M	5.1	21	21.8	28.9
kNN	3.3	14.2	20.5	23.2
MLP	5.6	19.7	30.2	40.6
RBF	1.75	10.88	10.99	11.93

Table 2. Confusion matrix obtained by RBF at experiments, level 3 of classification

	Homogeneous	Hole X	Hole Y	Crack XY	Crack ZY	Crack XZ	Multiple defects
Homogeneous	**1,000**	0,000	0,000	0,000	0,000	0,000	0,000
Hole X	0,000	**0,914**	0,000	0,000	0,048	0,000	0,038
Hole Y	0,000	0,000	**0,488**	0,107	0,298	0,107	0,000
Crack XY	0,000	0,000	0,022	**0,909**	0,009	0,025	0,034
Crack ZY	0,000	0,009	0,049	0,003	**0,930**	0,009	0,000
Crack XZ	0,000	0,000	0,005	0,003	0,044	**0,949**	0,000
Multiple defects	0,000	0,065	0,000	0,034	0,000	0,013	**0,888**

of six classes are well classified, except the "hole Y" class (48.8% success). This class is frequently confused with all the classes of cracks; it could be due to defect geometry does not allow produce a discernible wave pattern from propagation wave phenomena. In addition multiple-defect class is sometimes confused with cracks and hole X. It is due to particular patterns of one of the defects inside some multiple-defect specimens are more dominant in the spectra, causing multiple-defect spectra be alike to crack or hole Y spectra.

FUTURE TRENDS

The problem of material evaluation defined different levels of classification in a hierarchical outline with different kind of insight on quality of the tested material. It could be considered restate the problem to classify defects by ranges of defect size, independently of its shape or orientation, this kind of classification is very useful in industries as marble factories. The applicability of the proposed methodology has to be confirmed with application on different materials.

RBF neural network yielded good results for all levels of classification, but more algorithms have to be tested, taking into account the feasibility of its implementation in a real-time application and the improvement of the classification percentage of success. For instance, new algorithms of classification exploit linear dependencies in the data, and allow semi-supervised learning (Salazar, Vergara, Igual, Gosalbez, & Miralles, 2004), (Salazar, Vergara, Igual, & Gosalbez, 2005),

(Salazar, Vergara, Igual, & Serrano, 2007). That kind of modelling and learning procedure could be suitable for the classification of materials tested by impact-echo. Training stage and percentage of supervision is a critical subject in order to develop a suitable model from the data for classification. Thus depending on the kind of defective materials used in training a better adapted model for a specific classification would be defined. Then a decision fusion made by various classifiers could be more suitable than the decision made by one classifier.

CONCLUSION

We demonstrate the feasibility of using neural networks to extract patterns of different kinds of defects from impact-echo signal spectra in lab experiments. General results of the applied neural networks show RBF as the more suitable technique for the impact-echo problem even in complex levels of classifications, discerning up to 12 classes of homogeneous, one-defective and multiple-defect materials.

The proposed methodology has yield encouraging results with controlled lab experiments (same dimensions of the specimens, good-wave propagation material, and well-defined defects). The procedure has to be tested for processing real industry materials with a range of different dimensions, kind of defects and microstructures for which impact-echo signal spectra define fuzzy regions for classification.

REFERENCES

Abraham O, Leonard C., Cote P., & Piwakowski B. (2000). Time-frequency Analysis of Impact_Echo Signals: Numerical Modelling and Experimental Validation. *ACI Materials Journal, 97,* 647-657.

Bailer-Jones, C. (1996). *Neural Network Classifiation of Stellar Spectra.* University of Cambridge.

Bailer-Jones, C., Irwin, M., & Hippel, T. (1998). Automated classification of stellar spectra - II. Two-dimensional classification with neural networks and principal components analysis. *Monthly Notices of the Royal Astronomical Society, 298,* 361-377.

Bishop C.M. (2004). *Neural newtworks for pattern recognition.* Oxford: Oxford University Press.

Carino, N. J. (2001). The impact-echo method: an overview. In Structures Congress and Exposition (Ed.), (pp. 1-18).

Cheeke J.D. (2002). *Fundamentals and Applications of Ultrasonic Waves.* USA: CRC Press LLC.

Duda, R., Hart, P. E., & Stork, D. G. (2000). *Pattern classification.* (2 ed.) New York: Wiley-Interscience .

Hill, M., McHung, J., & Turner, J. D. (2000). Cross-sectional modes in impact-echo testing of concrete structures. *Journal-of-Structural-Engineering, 126,* 228-234.

Mei, X. (2004). Neural network for rapid depth evaluation of shallow cracks in asphalt pavements. *Computer-aided civil and infrastructure engineering, 19,* 223-230.

Pratt, D. & Sansalone, M. (1992). Impact-echo signal interpretation using artificial intelligence. *ACI Materials Journal, 89,* 178-187.

Salazar, A., Unió, J., Serrano, A., & Gosalbez, J. (2007). Neural Networks for Defect Detection in Non-Destructive Evaluation by Sonic Signals. *Lecture Notes in Computer Science, 4507,* 638-645.

Salazar, A., Vergara, L., Igual, J., & Gosalbez, J. (2005). Blind source separation for classification and detection of flaws in impact-echo testing. *Mechanical Systems and Signal Processing, 19,* 1312-1325.

Salazar, A., Vergara, L., Igual, J., Gosalbez, J., & Miralles, R. (2004). ICA Model Applied to Multichannel Non-destructive Evaluation by Impact-echo. *Lecture Notes in Computer Science, 3195,* 470-477.

Salazar, A., Vergara, L., Igual, J., & Serrano, A. (2007). Learning Hierarchies from ICA Mixtures. In I. 2. 20th International Joint Conference on Neural Networks (Ed.).

Sansalone, M., Carino, N. J., & Hsu, N. N. (1998). Transient stress waves interaction with planar flaws. *Batim-Int-Build-Res-Pract, 16,* 18-24.

Sansalone, M., Lin, Y., & Street, W. (1998). Determining the depth of surface-opening cracks using impact-generated stress waves and time-of-flight techniques. *ACI-Materials-Journal, 95,* 168-177.

Sansalone, M. & Street, W. (1997). *Impact-echo: Non-destructive evaluation of concrete and masonry.* New York: Bullbrier Press.

Stavroulakis, G. E. (1999). Impact-echo from a unilateral interlayer crack. LCP-BEM modelling and neural identification. *Engineering Fracture Mechanics, 62,* 165-184.

Xiang, Y. & Tso, S. K. (2002). Detection and classification of flaws in concrete structure using bispectra and neural networks. *NDT&E International, 35,* 19-27.

Xu, R., Nguyen, H., Sobol, P., Wang, S. L., Wu, A., & Johnson, K. E. (2004). Application of Principal Component Analysis to the FTIR Spectra of Disk Lubricant to Study Lube–Carbon Interactions. *IEEE Transactions on Magnetics, 40,* 3186-3189.

KEY TERMS

Accelerometer: A device that measures acceleration which is converted into an electrical signal that is transmitted to signal acquisition equipment. In impact-echo testing, the measured acceleration refers to vibration displacements caused by the excitation of the short impact.

Dimensionality Reduction: A process to reduce the number of variables of a problem. Dimension of a problem is given by the number of variables (features or parameters) that represent the data. After signal feature extraction (that reduce the original signal sample

space), the dimensionality may be reduced more by feature selection methods.

Fast Fourier Transform (FFT): A class of algorithms used in digital signal processing to compute the Discrete Fourier Transform (DFT) and its inverse. It has the capability of taking functions from the time domain to the frequency domain. The frequency components obtained are the spectra of the signal.

Feature Extraction (FE): A process to map a multidimensional space into a space of fewer dimensions. In signal processing, instead of processing raw signals with thousands of samples is more efficient to process features extracted from the signals, such as, signal power, principal frequency, and attenuation coefficient.

Feature Selection (FS): A technique that selects a subset of features from a given set of features that represent the relevant properties of the data. FS also may be define as the task of choosing a small subset of features which is sufficient to predict the target labels well, is crucial for efficient learning. There are several FS methods based on margins (e.g., relief, simba) or information theory (e.g., infogain). Supervised FS methods use a priori knowledge on a classification variable, to select variables high correlated with the known variable.

Leave-One-Out: A method used in classification with the following steps: i.) Label the database cases with the known classes. ii.) Select a case of the database. iii.) Estimate the class for selected case by a classifier using the remaining cases as training data. iv.) Repeat steps ii and iii until the end of the cases. v.) Calculate the mean percentage of success for classification results.

Signal Conditioner (SC): A device that converts one type of electronic signal into another type of signal. Its primary use is to convert a signal that may be difficult to read by conventional instrumentation into a more easily read format. Typical SC functions are amplification, electrical isolation, and linearization.

AVI of Surface Flaws on Manufactures I

Girolamo Fornarelli
Politecnico di Bari, Italy

Antonio Giaquinto
Politecnico di Bari, Italy

INTRODUCTION

The defect detection on *manufactures* is of utmost importance in the optimization of industrial processes (Garcia 2005). In fact, the industrial *inspection* of engineering materials and products tends to the detection, localization and classification of *flaws* as quickly and as accurately as possible in order to improve the production quality. In this field a relevant area is constituted by visual *inspection*. Nowadays, this task is often carried out by a human expert. Nevertheless, such kind of *inspection* could reveal time-consuming and suffer of low repeatability because the judgment criteria can differ from operator to operator. Furthermore, visual fatigue or loss of concentration inevitably lead to missed defects (Han, Yue & Yu 1999, Kwak, Ventura & Tofang-Sazi 2000, Y.A. Karayiannis, R. Stojanovic, P. Mitropoulos, C.Koulamas, T. Stouraitis, S. Koubias & G. Papadopoulos 1999, Patil, Biradar & Jadhav 2005).

In order to reduce the burden of human testers and improve the detection of faulty products, recently many researchers have been engaged in developing systems in Automated Visual *Inspection* (AVI) of *manufactures* (Chang, Lin & Jeng 2005, Lei 2004, Yang, Pang & Yung 2004). These systems reveal easily reliable from technical point of view and mimic the experts in the evaluation process of defects appropriately (Bahlmann, Heidemann & Ritter 1999), even if defect detection in visual *inspection* can become a hard task. In fact, in industrial processes a large amount of data has to be handled and *flaws* belong to a great number of classes with dynamic defect populations, because defects could present similar characteristics among different classes and different interclass features (R. Stojanovic, P. Mitropulos, C.Koullamas, Y. Karayiannis, S. Koubias & G. Papadopoulos 2001). Therefore, it is needed that visual *inspection* systems are able to adapt to dynamic operating conditions. To this purpose *soft computing* *techniques* based on the use of Artificial Neural Networks (ANNs) have already been proposed in several different areas of industrial production. In fact, neural networks are often exploited for their ability to recognize a wide spread of different defects (Kumar 2003, Chang, Lin & Jeng 2005, Garcia 2005, Graham, Maas, Donaldson & Carr 2004, Acciani, Brunetti & Fornarelli 2006). Although adequate in many instances, in other cases Neural Networks cannot represent the most suitable solution. In fact, the design of ANNs often requires the extraction of parameters and features, during a *preprocessing* stage, from a suitable data set, in which the most possible defects are recognized (Bahlmann, Heidemann & Ritter 1999, Karras 2003, Rimac-Drlje, Keller & Hocenski 2005). Therefore, methods based on neural networks could be time expensive for in-line applications because such preliminary steps and could reveal complex (Kumar 2003, Kwak, Ventura & Tofang-Sazi 2000, Patil, Biradar & Jadhav 2005, R. Stojanovic, P. Mitropulos, C.Koullamas, Y. Karayiannis, S. Koubias & G .Papadopoulos 2001). For this reason, when in an industrial process time constraints play an important role, a hardware solution of the abovementioned methods can be proposed (R. Stojanovic, P. Mitropulos, C.Koullamas, Y. Karayiannis, S. Koubias & G .Papadopoulos 2001), but such kind of solution implies a further design effort which can be avoided by considering Cellular Neural Networks (CNNs) (Chua & Roska 2002).

Cellular Neural Networks have good potentiality to overcome this problem, in fact their hardware implementation and massive parallelism can satisfy urgent time constrains of some industrial processes, allowing the inclusion of the diagnosis inside the production process. In this way the defect detection method could enable to work in *real time* according to the specific industrial process.

BACKGROUND

Cellular Neural Networks consist of processing units $C(i, j)$, which are arranged in an $M \times N$ grid, as shown in Figure 1.

The generic basic unit $C(i, j)$ is called cell: it corresponds to a first-order nonlinear circuit, electrically connected to the cells, which belong to the set $S_r(i, j)$, named *sphere of influence of the radius r* of $C(i, j)$. Such set $S_r(i, j)$ is defined as:

$$S_r(i,j) = \left\{ C(k,l) \Big|_{1 \leq k \leq M, 1 \leq l \leq N} \max \left(|k-i|, |l-j| \right) \leq r \right\}$$

An $M \times N$ Cellular Neural Network is defined by an $M \times N$ rectangular array of cells $C(i, j)$ located at site (i, j), $i = 1, 2, .., M, j = 1, 2, …, N$. Each cell $C(i, j)$ is defined mathematically by the following state and output equations:

$$\begin{cases} \dfrac{dx_{ij}}{dt} = -x_{ij} + \sum_{C(k,l) \in S_r(i,j)} A(i,j;k,l)y_{kl} + \sum_{C(k,l) \in S_r(i,j)} B(i,j;k,l)u_{kl} + z_{ij} \\ y_{ij} = \dfrac{1}{2}\left(|x_{ij}+1| - |x_{ij}-1| \right) \end{cases}$$

where $x_{ij} \in \Re$, $y_{ij} \in \Re$ and $z_{ij} \in \Re$ are state, output and threshold of cell $C(i, j)$, $y_{kl} \in \Re$, and $u_{kl} \in \Re$ are output and input of cell $C(k, l)$, respectively. $A(i, j; k, l)$ and

Figure 1. Standard CNN architecture

$B(i, j; k, l)$ are called the *feedback* and the *input synaptic operators* and uniquely identify the network.

The reported circuit model constitutes a hardware paradigm which allows fast processing of signals. For this reason, in the past CNNs were considered as an useful framework for defect detection in industrial applications (Roska 1992). Successively different CNN-based contributions working in *real time* and aiming at the defect detection in the industrial field have been proposed (Bertucco, Fargione, Nunnari & Risitano 2000), (Occhipinti, Spoto, Branciforte & Doddo 2001), (Guinea, Gordaliza, Vicente & García-Alegre 2000), (Perfetti & Terzoli 2000). In (Bertucco, Fargione, Nunnari & Risitano 2000) and (Occhipinti, Spoto, Branciforte & Doddo 2001) non-destructive control of mechanical parts in aeronautical industrial production is carried out defining an algorithm which is implemented by means of CNNs entirely. These methods reveal effective, but a complex acquisition system is required to provide information about the defectiveness. In (Guinea, Gordaliza, Vicente & García-Alegre 2000) CNNs constitute the core processors of a system which realizes an automatic *inspection* of metal laminates, whereas in (Perfetti & Terzoli 2000) two CNN-based algorithms are proposed in order to detect stains and irregularities in a textile application. In both works real-time is guaranteed, but in (Guinea, Gordaliza, Vicente & García-Alegre 2000) synthesis criteria of CNN circuit parameters could reveal difficult to satisfy, whereas in (Perfetti & Terzoli 2000) such criteria are not defined.

In the following section a CNN-based method, that enables to overcome the most of drawbacks which arise in the reported approaches, is proposed.
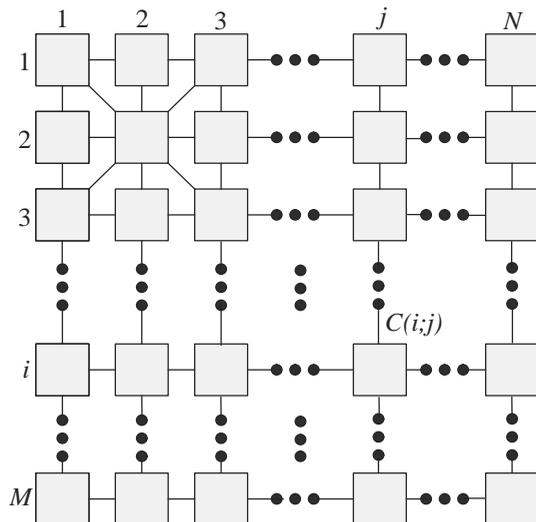
AUTOMATIC DEFECT DETECTION METHOD

In this section an automatic method for the visual *inspection* of surface *flaws* of *manufactures* is proposed. This method is realized by means of a CNN-based architecture, which will be accurately described in the companion chapter (Fornarelli & Giaquinto 2007).

The suggested approach consists of three steps. The first one realizes a *preprocessing* stage which enables to identify eventual defected areas; in the second stage the matching between such pre-processed image and a *reference image* is performed; finally, in the third

A

step an output binary image, in which only defects are represented, is yielded.

The proposed solution needs nor complex acquisition system neither feature extraction, in fact the image is directly processed and the synthesis parameters of the system are evaluated from the statistical image properties automatically. Furthermore, the proposed system is well suited for single board implementation.

The scheme that represents the proposed method is shown in Figure 2.

As it can be observed, it is formed by three modules: a *Preprocessing* module, an Image Matching module and a Defect Detection one. The input images, named **O** and **R**, are acquired by means of a camera, which yields 256-gray levels images, whose dimensions are $m \times n$. The image **O** represents the *manufacture* under test or a part of it. Such image contains the Region of Interest (ROI), that is the specific region of an object, in which defects are to be detected.

The image **R** constitutes a *reference image*, in which a product without defects (or its part) is depicted. Such image is stored in a memory and acquired off-line during the phase of system calibration. It is used to detect possible variations caused by the presence of dents, scratches or breakings on an observed surface. In order to allow a good match between the *reference image* and the under test one, the *preprocessing* blocks realize a *contrast enhancement*, providing images O_F and R_F, that constitute the inputs for the subsequent

Image Matching module. The target of this block is finding the minimum difference between the two images O_F and R_F. In fact, during the production process, the acquiring system could give images in which the *manufacture* is shifted according the four cardinal directions. This implies that the difference between O_F and R_F could lead to the detection of false defects. The Image Matching Module minimizes such effects, looking for the best matching between the image to be processed and the reference one. Successively the difference image **D** feeds the Defect Detection module. This part aims at the detection of the presence of *flaws* on the product under test and gives an output image containing only the defects. The output image allows to activate alarming systems able to detect the presence of *flaws*, making this industrial task easier, in fact it could support experts in their diagnoses. The detailed implementation of each module will be illustrated in the second part of this contribution.

FUTURE TRENDS

In order to provide the most information related to defects detected by the proposed approach in industrial processes, features of *flaws* should be identified. For this reason, future works will be devoted to the evaluation of different characteristics like dimension of defects, kind of damage and its degree. Moreover, the advantages by applying the proposed method in various industrial fields will be investigated and techniques minimizing eventual misclassifications in particular applications, will be developed.

Figure 2. Block diagram of the proposed CNN-based method

CONCLUSION

In this chapter a CNN-based method for the visual *inspection* of surface *flaws* of *manufactures* has been proposed. The approach consists of three modules: a *Preprocessing* Module provides images, in which contrast is enhanced. An Image Matching Module allows to make up for eventual misalignment between the *manufacture* under test and the acquisition system. Finally, the Defect Detection Module enables to extract images which contain defects of *manufactures*.

The suggested method offers attractive advantages. It reveals general, therefore it can be introduced in different industrial fields, in which the identification

of superficial anomalies like dents, corrosions or spots on *manufactures* is a fundamental task.

Moreover, the suggested method is finalized to the implementation by means of an architecture, entirely formed by Cellular Neural Networks, exploiting the potentialities that this kind of network offers in processing signals. Therefore, the proposed approach enables to automate in line diagnosis processes reducing operators' burden in identifying production defects.

REFERENCES

G. Acciani, G. Brunetti & G. Fornarelli (2006), "Application of Neural Networks in Optical Inspection and Classification of Solder Joints in Surface Mount Technology", *IEEE Transactions on Industrial Informatics*, vol. 2, no. 3, ISSN 1551-3203, pp. 200-209.

C. Bahlmann, G. Heidemann & H. Ritter (1999), "Artificial neural networks for automated quality control of textile seams", *Pattern Recognition,* vol. 32, pp. 1049-1060.

L. Bertucco, G. Fargione, G. Nunnari & A. Risitano (2000), "A Cellular Neural Network Approach for Nondestructive Control of Mechanical Parts" *Proc. of the 6th IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, Catania, Italy, 23-27 May 2000, pp. 159-164.

C.Y. Chang, S.Y. Lin & M.D. Jeng (2005), " Using a Two-layer Competitive Hopfield Neural Network for Semiconductor Wafer Defect Detection", *Proc. of the 2005 IEEE International Conference on Automation, Science and Engineering,* August 1-2, 2005, pp. 301-306.

L.O. Chua & T. Roska (2002), "Cellular Neural Networks and Visual Computing: Foundation and Applications", *Cambridge University Press,* Cambridge, United Kingdom, 2002.

G. Fornarelli & A. Giaquinto (2007), "A CNN-based Technique for the Identification of Superficial Anomalies on Manufactures – Part II", submitted to *Encyclopedia of Artificial Intelligence,* Editors: J.R. Rabuñal, J. Dorado & A. Pazos, Information Science Reference, 2007.

C. Garcia (2005), "Artificial intelligence applied to automatic supervision, diagnosis and control in sheet metal stamping processes", *Journal of Material Processing Technology*, vol. 164-165, pp. 1351-1357.

D. Graham, P. Maas, G.B. Donaldson & C. Carr (2004), "Impact damage detection in carbon fibre composites using HTS SQUIDs and neural networks", *NDT&E International*, vol 37, pp. 565-570.

D. Guinea, A. Gordaliza, J. Vicente & M.C. García-Alegre (2000), "CNN based visual processing for industrial inspection" *Proc. of The International Society for Optical Engineering (SPIE)*, vol. 3966, no. 45, 2000, pp. p. 315-322.

L. Han, X. Yue & J. Yu (1999), "Inspection of Surface Defects on Engine Cylinder Wall Using Segmentation Image Processing Technique", *Proc. of the IEEE International Conference on Vehicle Electronics,* 6-9 September, 1999, Changchun, P.R. China, vol. 1, pp. 258-260.

Y.A. Karayiannis, R. Stojanovic, P. Mitropoulos, C.Koulamas, T. Stouraitis, S. Koubias & G. Papadopoulos (1999), "Defect Detection and Classification on Web Textile Fabric using Multiresolution Decomposition and Neural Networks", *Proc. Of the 6th IEEE International Conference on Electronics, Circuits and Systems*, 5-8 September 1999, Pafos, Cyprus, pp. 765-768.

D.A. Karras (2003), " Improved Defect Detection Using Support Vector Machines and Wavelet Feature Extraction Based on Vector Quantization and SVD Techniques", *Proc. of the International Joint Conference on Neural Networks*, 20-24 July 2003, Portland, Oregon, USA, pp. 2322-2327

A. Kumar (2003), " Neural network based detection of local textile defects ", *Pattern Recognition,* vol. 36, pp. 1645-1659.

C. Kwak, J.A. Ventura & K. Tofang-Sazi (2000), "A neural network approach for defect identification and classification on leather fabric", *Journal of Intelligent Manufacturing,* vol. 11, pp. 485-499.

L. Lei (2004), "A Machine Vision System for Inspecting Bearing-Diameter", *Proc. of the IEEE 5th World Congress on Intelligent Control and Automation*, 15-19 June 2004, Hangzhou P.R. China, pp. 3904-3906.

A

L. Occhipinti, G. Spoto, M. Branciforte & F. Doddo (2001), "Defects Detection and Characterization by Using Cellular Neural Networks"**,** *Proc. of the 2001 IEEE International Symposium on Circuits and Systems, (ISCAS 2001)*, 6-9 May 2001, vol. 3, pp. 481-484.

P.M. Patil, M.M. Biradar & S. Jadhav (2005), "Orientated texture segmentation for detecting defects", *Proc. of the International Joint Conference on Neural Networks,* 31 July-4 August 2005, Montreal, Canada, pp. 2001-2005.

R. Perfetti & L. Terzoli (2000), "Analogic CNN algorithms for textile applications", *International Journal of Circuit Theory and Applications*, no. 28, pp. 77-85.

S. Rimac-Drlje, A. Keller & Z. Hocenski (2005), "Neural Network Based Detection of Defects in Texture Surfaces", *IEEE International Symposium on Industrial Electronics*, 20-23 June 2005, Dubrovnik, Croatia, pp. 1255-1260.

T. Roska (1992), "Programmable CNN: a Hardware Accelerator for Simulation, Learning, and Real-Time Applications", *Proc. of the 35th Midwest Symposium*, Washington, USA, August 9-12, 1992, vol.1 pp. 437-440.

R. Stojanovic, P. Mitropulos, C. Koullamas, Y. Karayiannis, S. Koubias, G .Papadopoulos (2001), "Real-Time Vision-Based System for Textile Fabric Inspection", *Real-Time Imaging,* vol. 7, pp. 507-518.

X. Yang, G. Pang, N. Yung (2004), "Discriminative training approaches to fabric defect classification based on wavelet transform", *Pattern Recognition*, vol. 37, pp. 889-899.

KEY TERMS

Artificial Neural Networks: A set of basic processing units which communicate to each other by weighted connections. These units give rise a parallel processing with particular properties such as the ability to adapt or learn, to generalise, to cluster or organise data, to approximate non-linear functions. Each unit receives an input from neighbours or external sources and uses it to compute an output signal. Such signal is propagated to other units or is a component of the network output. In order to map an input set into an output one a neural network is trained by teaching patterns, changing its weights according to proper learning rules.

Automated Visual Inspection: An automatic form of quality control normally achieved using one or more cameras connected to a processing unit. Automated Visual Inspection has been applied to a wide range of products. Its target consists of minimizing the effects of visual fatigue of human operators who perform the defect detection in a production line environment.

Cellular Neural Networks: A particular circuit architecture which possesses some key features of Artificial Neural Networks. Its processing units are arranged in an M×N grid. The basic unit of Cellular Neural Networks is called cell and contains linear and non linear circuit elements. Each cell is connected only to its neighbour cells. The adjacent cells can interact directly with each other, whereas cells not directly connected together may affect each other indirectly because of the propagation effects of the continuous time dynamics.

Defect Detection: Extraction of information about the presence of an instance in which a requirement is not satisfied in industrial processes. The aim of Defect Detection consists of highlighting manufactures which are incorrect or missing functionality or specifications.

Image Matching: Establishment of the correspondence between each pair of visible homologous image points on a given pair of images, aiming at the evaluation of novelties.

Industrial Inspection: Analysis pursuing the prevention of unsatisfactory industrial products from reaching the customer, particularly in situations where failed manufactures can cause injury or even endanger life.

Region of Interest: A selected subset of samples within a dataset identified for a particular purpose. In image processing, the Region of Interest is identified by the boundaries of an object. The encoding of a Region of Interest can be achieved by basing its choice on: (a) a value that may or may not be outside the normal range of occurring values; (b) purely separated graphic information, like drawing elements; (c) separated semantic information, such as a set of spatial and/or temporal coordinates.

AVI of Surface Flaws on Manufactures II

Girolamo Fornarelli
Politecnico di Bari, Italy

Antonio Giaquinto
Politecnico di Bari, Italy

INTRODUCTION

Automatic visual inspection takes a relevant place in *defect detection* of industrial *production*. In this field a fundamental role is played by methods for the detection of superficial anomalies on manufactures.

In particular, several systems have been proposed in order to reduce the burden of human operators, avoiding the drawbacks due to the subjectivity of judgement criteria (Kwak, Ventura & Tofang-Sazi 2000, Patil, Biradar & Jadhav 2005).

Proposed solutions are required to be able to handle and process a large amount of data. For this reason, neural networks-based methods have been suggested for their ability to deal with a wide spread of data (Kumar 2003, Chang, Lin & Jeng 2005, Garcia 2005, Graham, Maas, Donaldson & Carr 2004, Acciani, Brunetti & Fornarelli 2006). Moreover, in many cases these methods must satisfy time constrains of industrial processes, because the inclusion of the diagnosis inside the *production* process is needed.

To this purpose, architectures, based on Cellular Neural Networks (CNNs), revealed successful in the field of *real time defect detection*, due to the fact that these networks guarantee a hardware *implementation* and massive parallelism (Bertucco, Fargione, Nunnari & Risitano 2000), (Occhipinti, Spoto, Branciforte & Doddo 2001), (Perfetti & Terzoli 2000). On the basis of these considerations, a method to identify superficial damages and anomalies in manufactures has been given in (Fornarelli & Giaquinto 2007). This method is aimed at the *implementation* by means of an architecture entirely formed by Cellular Neural Networks, whose synthesis is illustrated in the present work. The suggested solution reveals effective for the detection of defects, as shown by two test cases carried out on an injection pump and a sample textile.

BACKGROUND

In the companion paper an approach for *defect detection* of surface flaws on manufactures is proposed: this approach can be divided into three modules, named Preprocessing module, Image Matching module and *Defect Detection* module, respectively. The first one realizes a pre-processing stage which enables to identify eventual defected areas; in the second stage the *matching* between such pre-processed image and a reference one is performed; finally, in the third step an output binary image, in which only defects are represented, is yielded.

The proposed solution needs nor complex acquisition system neither feature extraction, in fact the image is directly processed and the synthesis parameters of the networks are evaluated from the statistical image properties automatically. Furthermore, the proposed system is well suited for a single board *implementation*.

CNN-BASED DIAGNOSIS ARCHITECTURE

The detailed *implementation* of each module will be illustrated in the following. Successively the results obtained by testing the suggested architecture on two real cases are shown and a discussion of numerical outcomes is reported.

Preprocessing Module

The Preprocessing is realized by a *Fuzzy Contrast Enhancement* block. This block consists of a *Fuzzy* Associative Memory (FAM), developed as the preprocessing stage of the CNN-based system considered in (Carnimeo & Giaquinto 2002). The proposed circuit enables to transform 256-gray levels images into fuzzified ones, whose contrast is enhanced, due to the fact

that their *histogram*s are stretched. To this purpose a proper fuzzification procedure is developed to define two *fuzzy* subsets adequate to describe the semantic content of patterns such as images of industrial objects, which can be classified as belonging to the **Object/Background** class.

In an analogous way, the domain of output values has been characterized by means of two output *fuzzy* subsets defined as **Dark** and **Light**. In particular, the *fuzzy* rules which provide the mapping from original images (**O/R**) into fuzzified ones (**O$_F$/R$_F$**) can be expressed as:

IF O(i, j) $\in$ **Object**   THEN O$_F$(i, j) $\in$ **Dark**
IF O(i, j) $\in$ **Background** THEN O$_F$(i, j) $\in$ **Light**

where O(i, j) and O$_F$(i, j) denote the gray level value of the (i, j)-th pixel in the original image and in the fuzzified one, respectively. As showed in (Carnimeo & Giaquinto 2002), the reported *fuzzy* rules can be encoded into a single FAM.

Then, a Cellular Neural Network is synthesized to behave as the codified FAM by adopting the synthesis procedure developed in (Carnimeo & Giaquinto 2002), where the synthesis of a CNN-based memory, which contains the abovementioned fuzzification rules is accurately formulated.

Contrasted images present a stretched *histogram*. This implies that such operation minimizes the effects of image noise, caused by environmental problems like dust or dirtiness of camera lenses. Moreover, it reduces the undesired information due to the combination between the non uniformity of the illumination in the image and the texture of the manufacture (Jamil, Bakar, Mohd, & Sembok 2004).

Image Matching Module

In Figure 1 the block diagram corresponding to the Image Matching module is reported. The target of this module consists of finding the best *matching* between the images yielded by processing the acquired image and the reference one. To this purpose the image **O$_F$** is shifted by one pixel into the four cardinal directions (NORTH; SOUTH, EAST and WEST), using four space-invariant CNNs (T. Roska, L. Kek, L. Nemes, A. Zarandy & P. Szolgay 1999) and obtaining the images **O$_{FN}$**, **O$_{FS}$**, **O$_{FE}$** and **O$_{FW}$**. Successively the switch S$_1$ changes its position, excluding the image **O$_F$**. The reference image **R$_F$** is subtracted by the images **O$_{FN}$**, **O$_{FS}$**, **O$_{FE}$** **O$_{FW}$** and **O$_F$**, then the number b$_N$, b$_S$, b$_E$, b$_W$ and b$_0$ of black pixels in the resulting images **D$_N$**, **D$_S$**, **D$_E$**, **D$_W$** and **D$_0$** are computed. The image, which best matches with the reference one, presents the maximum numbers of black pixels. Therefore, such value drives the switch S$_2$, which allows to feedback the image that best matches with the reference one. In this way the image which presents the minimum difference becomes the input for a successive computational step. The processing is repeated until **D$_0$** presents the best *matching*. When this condition is satisfied, the difference image **D** between **D$_0$** and **R$_F$** is computed. As it can be noticed the operations needed for each directional shift can be carried on simultaneously, reducing the computational time at each step.

Defect Detection Module

The third part of the suggested architecture is a *Defect Detection* module. The subsystem is synthesized with the aim of computing the output binary image **F**, in which only the defects are present. Such module is composed by the sequence of a Major Voting circuit, a CNN associative memory for *contrast enhancement* and a Threshold circuit. The corresponding CNN-based *implementation* is obtained by considering space invariant networks.

In detail the Major Voting circuit minimizes the number of false detections caused by the presence of noise, highlighting the dents or those flaws which lead to a changing in the reflectance of light in the original image. The output of the Major Voting block **D$_M$** feeds a CNN working as the associative memory described in the previous Preprocessing module subsection. This operation provides an output image **D$_{MF}$** whose *histogram* is bimodal. In this kind of image the selection of a threshold, which highlights the defects, results feasible. In fact, a proper value is given by the mean of the modes of the *histogram*. Then, this image is segmented by means of the corresponding space-invariant CNN (T. Roska, L. Kek, L. Nemes, A. Zarandy & P. Szolgay 1999), obtaining the corresponding binary image **F**. In this way errors corresponding to incorrect identification of defects are minimized because only flaws are visible after the *segmentation*.

Figure 1. Block diagram corresponding to the Image Matching module
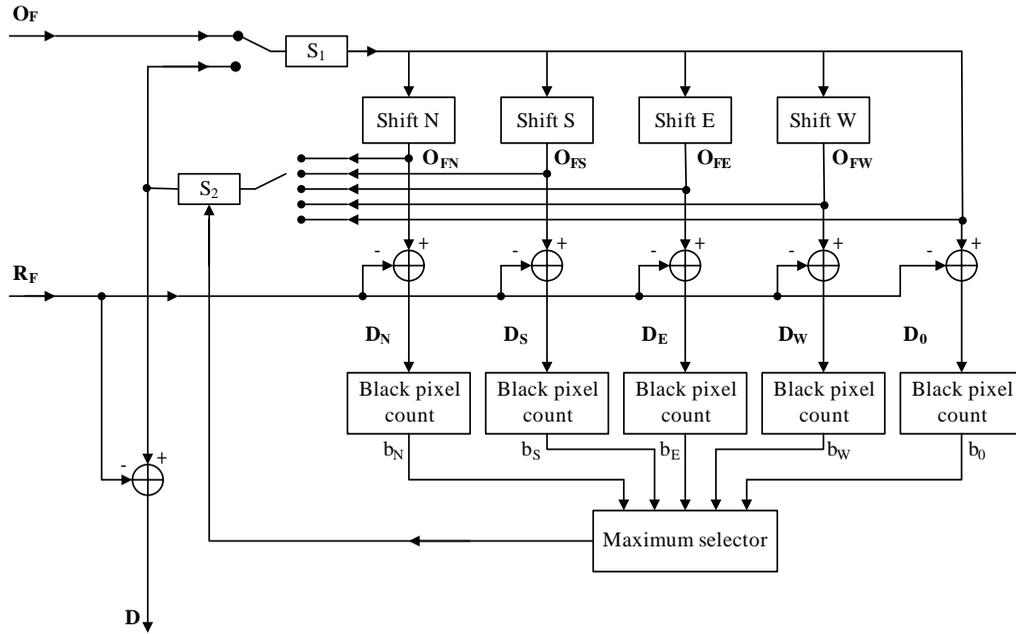
Numerical Examples

The capabilities of the designed CNN-based architecture have been investigated on images representing the central part of injection pumps containing the Region of Interest (ROI), that is the flange, like the reported in Figure 2(a), whose *histogram* is shown in Figure 2(b). As it can be observed, this image presents two dents on the left and the bottom of the observed region. Dents are due to the collisions that can occur when pumps are moved among the different *production* locations during the various stages of the mounting. This image and the reference one are firstly processed by a circuit based on two (4×4)-cell CNNs described in the previous subsection. In Figures 3(a-b) the corresponding output image yielded by the synthesized CNNs and its *histogram* are shown. It can be noticed that contrast is highly enhanced and the *histogram* is stretched.

In figure 4(a) the output **D** of the Image Matching module is reported: impulsive-like noise due to the shifting or the imperfect lighting of the image or the reflection due to dirtiness is still present at this step.

Finally, **D** feeds the *Defect Detection* module: in Figures 4(b) and 4(c) the output of Major Voting block **D**$_M$ and the final image **F** are shown, respectively. As it can be observed in image **D**$_M$, the effects of irregular lighting or changing in reflections due to the dust or

dirtiness are minimized. The results are encouraging, in fact the designed cellular system provides an output image (see Figure 4(c)), in which the areas of the manufacture with defects are well visible and detected by white pixels.

Performances of the proposed system have been tested by means of a second experiment carried out on a sample textile. This industrial field has been investigated because time constraints of an automated *defect detection* system in the textile industry are of crucial importance (R. Stojanovic, P. Mitropulos, C.Koullamas, Y. Karayiannis, S. Koubias & G.Papadopoulos 2001).

In Figure 5 (a) the acquired image of the textile is reported. In this case the whole image of the manufacture coincides with the ROI. It can be noted that a bright vertical thin bar compares in the middle of the image. It corresponds to a lacked stamp. In the reported example the identification of defects constitutes a non trivial problem. In fact, the stamped areas have variegated geometric shapes, which can be depicted with a great number of different gray levels. The previously reported method has been applied to detect such kind of defects, yielding an image in which the only defect (the thin bar) is represented, similarly to the test case reporting dents of injection pumps. In Figure 5(b)-(c)-(d) the corresponding outputs of the Preprocessing module, the Image *Matching* module and the *Defect*

213

Figure 2. (a) Acquired Image containing the ROI (the flange of injection pump) with two dents; (b) gray-scale histogram of the image in Figure 2(a)

(a)

(b)
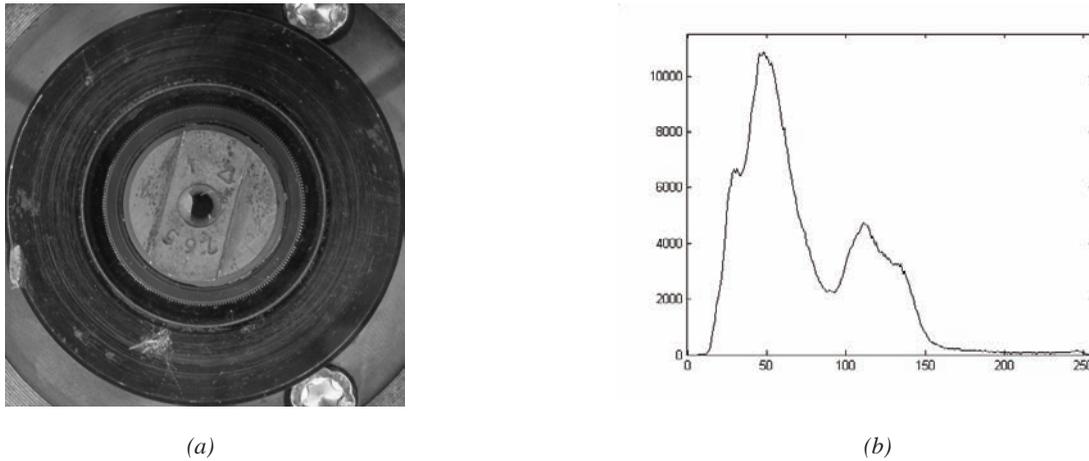
Figure 3. (a) Output image of the Fuzzy Contrast Enhancement module fed by the image in Figure 2(a); (b) gray-scale histogram of the image in Figure 3(a)

(a)

(b)
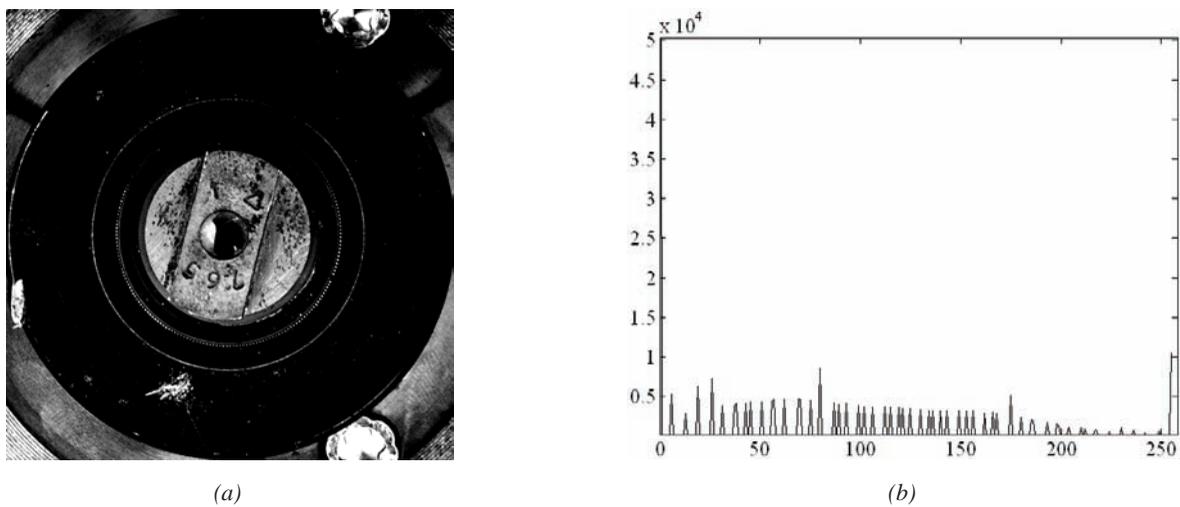
Detection one are shown, respectively. It can be noticed that the central defect has been isolated effectively, even if a percentage of the areas to be identified is missed. This is due to the fact that, when details need to be detected, it is required that contrast is maximum. Nevertheless, as the contrast is increased, the *histogram* of the resulting image is emphasized toward extreme values of gray levels with respect to the acquired image. This implies that, due to a saturation phenomenon,

an information loss about details takes place. (Brendel & Roska 2002).

Finally, in the output image small white areas are misclassified as defects. This problem rises from the shift of the manufacture respect to the acquisition system. The Image *Matching* module minimizes the effects of such problem, but it can not delete them completely when mechanical deformations of the manufacture occur as in the textile field. As it is shown in Figure 5(d),

Figure 4. (a) Output of the image matching module; (b) output of the major voting block in the defect detection module; (c) output image containing the detected defects, represented by white pixels

A

(a)   *(b)*   *(c)*
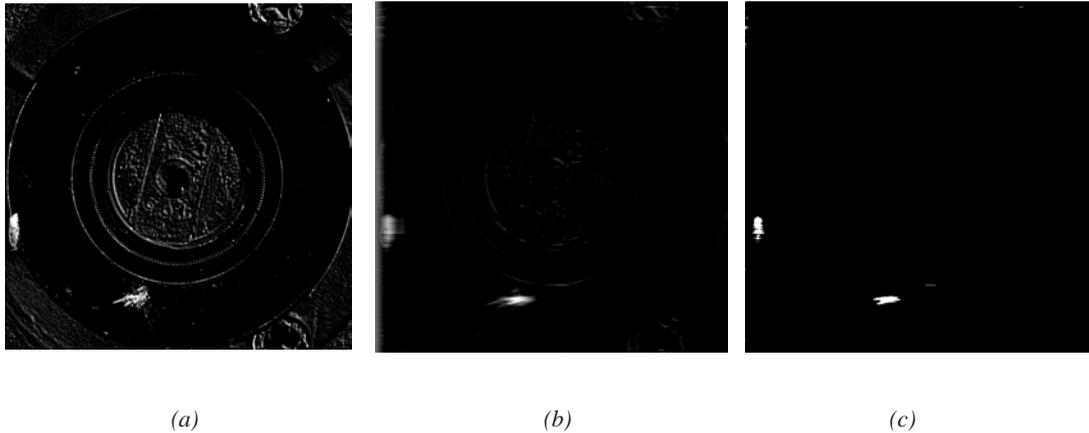
Figure 5. (a) Acquired image of a textile containing a thin bar, (b) corresponding output of the fuzzy contrast enhancement module; (c) output of the image matching module, (d) final output image

(a)   *(b)*   *(c)*   *(d)*
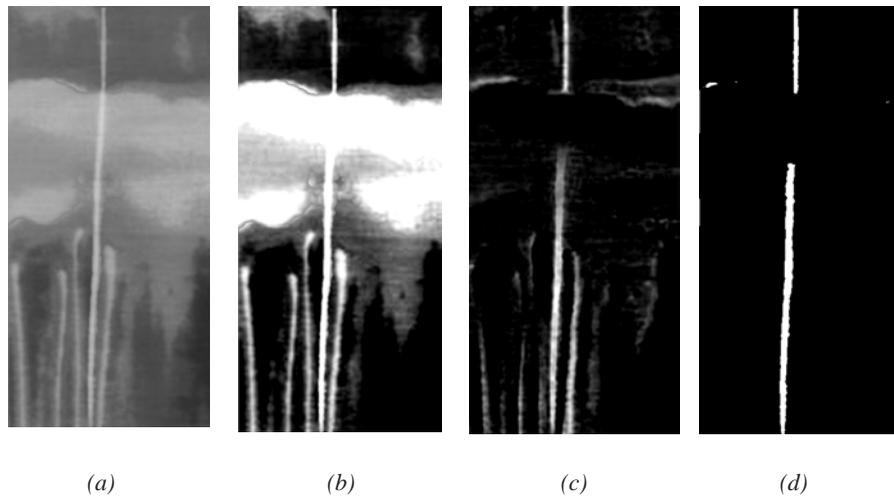
this implies the presence of false positives, which will be investigated subsequently.

FUTURE TRENDS

As it can be argued from an observation of obtained numerical results, future works will be devoted to a more detailed analysis of misclassifications. In particular,

false positives could be analyzed by means of further techniques which relate the characteristics of the possible defected zones and the ones containing effective defects according to the constraints of the application. For instance, in the reported numerical examples the false positives have geometric sizes which are negligible if compared to the areas of eventual flaws. Therefore, a control of area dimensions could enable to discriminate the two kinds of regions.

CONCLUSION

In this paper a CNN-based architecture for the visual inspection of surface flaws of manufactures has been proposed. The architecture consists of modules, which are entirely realized by well-established circuit networks. The reported design approach offers some interesting advantages. The proposed solution needs nor complex acquisition system neither feature extraction, in fact images are directly processed and the synthesis parameters, like thresholds for image *segmentation*, are evaluated from the statistical image properties automatically. Furthermore, due to the possible hardware *implementation* of CNNs the resulting system can satisfy urgent time constrains relating to the in-line detection of some industrial productive processes, allowing the inclusion of the diagnosis inside the *production* steps.

REFERENCES

G. Acciani, G. Brunetti & G. Fornarelli (2006), "Application of Neural Networks in Optical Inspection and Classification of Solder Joints in Surface Mount Technology", *IEEE Transactions on Industrial Informatics*, vol. 2, no. 3, ISSN 1551-3203, pp. 200-209.

L. Bertucco, G. Fargione, G. Nunnari & A. Risitano (2000), "A Cellular Neural Network Approach for Nondestructive Control of Mechanical Parts" *Proc. of the 6th IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, Catania, Italy, 23-27 May 2000, pp. 159-164.

M. Brendel & T. Roska (2002), "Adaptive Image Sensing and Enhancement Using the Cellular Neural Network Universal Machine", *International Journal of Circuit Theory and Applications*, vol. 30, Issue 2-3, pp. 287-312.

L. Carnimeo & A. Giaquinto (2002), "A Cellular Fuzzy Associative Memory for Bidimensional Pattern Segmentation", *Proc. of the 7th IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, Frankfurt, Germany, July 22-24, 2002, pp. 430-435.

C.Y. Chang, S.Y. Lin & M.D. Jeng (2005), " Using a Two-layer Competitive Hopfield Neural Network for Semiconductor Wafer Defect Detection", *Proc. of the 2005 IEEE International Conference on Automation, Science and Engineering,* August 1-2, 2005, pp. 301-306.

G. Fornarelli & A. Giaquinto (2007), "A CNN-based Technique for the Identification of Superficial Anomalies on Manufactures – Part I", submitted to *Encyclopedia of Artificial Intelligence,* Editors: J.R. Rabuñal, J. Dorado & A. Pazos, Information Science Reference, 2007.

C. Garcia (2005), "Artificial intelligence applied to automatic supervision, diagnosis and control in sheet metal stamping processes", *Journal of Material Processing Technology*, vol. 164-165, pp. 1351-1357.

D. Graham, P. Maas, G.B. Donaldson & C. Carr (2004), "Impact damage detection in carbon fibre composites using HTS SQUIDs and neural networks", *NDT&E International*, vol 37, pp. 565-570.

A. Kumar (2003), " Neural network based detection of local textile defects ", *Pattern Recognition,* vol. 36, pp. 1645-1659.

C. Kwak, J.A. Ventura & K. Tofang-Sazi (2000), "A neural network approach for defect identification and classification on leather fabric", *Journal of Intelligent Manufacturing,* vol. 11, pp. 485-499.

N. Jamil, Z. A. Bakar, T. Mohd & T. Sembok (2004), "A Comparison of Noise Removal Techniques in Songket Motif Images", *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV'04)*, 26-29Jul 2004, Penang, Malaysia, pp. 139-143.

L. Occhipinti, G. Spoto, M. Branciforte & F. Doddo (2001), "Defects Detection and Characterization by Using Cellular Neural Networks", *Proc. of the 2001 IEEE International Symposium on Circuits and Systems, (ISCAS 2001)*, 6-9 May 2001, vol. 3, pp. 481-484.

P.M. Patil, M.M. Biradar & S. Jadhav (2005), "Orientated texture segmentation for detecting defects", *Proc. of the International Joint Conference on Neural Networks,* 31July-4August 2005, Montreal, Canada, pp. 2001-2005.

R. Perfetti & L. Terzoli (2000), "Analogic CNN algorithms for textile applications", *International Journal of Circuit Theory and Applications*, no. 28, pp. 77–85.

T. Roska, L. Kek, L. Nemes, A. Zarandy & P. Szolgay (1999), "CSL CNN Software Library (Templates and Algorithms) Vers. 7.3", *Budapest, Hungary*.

A

R. Stojanovic, P. Mitropulos, C. Koullamas, Y. Karayiannis, S. Koubias, G. Papadopoulos (2001), "Real-Time Vision-Based System for Textile Fabric Inspection", *Real-Time Imaging,* vol. 7, pp. 507-518.

KEY TERMS

Automated Visual Inspection: An automatic form of quality control normally achieved using one or more cameras connected to a processing unit. Automated Visual Inspection has been applied to a wide range of products. Its target consists of minimizing the effects of visual fatigue of human operators who perform the defect detection in a production line environment.

Cellular Neural Networks: A particular circuit architecture which possesses some key features of Artificial Neural Networks. Its processing units are arranged in an M×N grid. The basic unit of Cellular Neural Networks is called cell and contains linear and non linear circuit elements. Each cell is connected only to its neighbour cells. The adjacent cells can interact directly with each other, whereas cells not directly connected together may affect each other indirectly because of the propagation effects of the continuous time dynamics.

Fuzzy Associative Memory: A kind of content-addressable memory in which the recall occurs correctly if input data fall within a specified window consisting of an upper bound and a lower bound of the stored patterns. A Fuzzy Associative Memory is identified by a matrix of fuzzy values. It allows to map an input fuzzy set into an output fuzzy one.

Histogram Stretching: A point process that involves the application of an appropriate transformation function to every pixel of a digital image in order to redistribute the information of the histogram toward the extremes of a grey level range. The target of this operation consists of enhancing the contrast of digital images.

Image Matching: Establishment of the correspondence between each pair of visible homologous image points on a given pair of images, aiming at the evaluation of novelties.

Major Voting: An operation aiming at deciding whether the neighbourhood of a pixel in a digital image contains more black or white pixels, or their number is equal. This effect is realized in two steps. The first one gives rise to an image, where the sign of the rightmost pixel corresponds to the dominant colour. During the second step the grey levels of the rightmost pixels are driven into black or white values, depending on the dominant colour, or they are left unchanged otherwise.

Real Time System: A system that must satisfy explicit bounded response time constraints to avoid failure. Equivalently, a real-time system is one whose logical correctness is based both on the correctness of the outputs and its timeliness. The timeliness constraints or deadlines are generally a reflection of the underlying physical process being controlled.

Region of Interest: A selected subset of samples within a dataset identified for a particular purpose. In image processing, the Region of Interest is identified by the boundaries of an object. The encoding of a Region of Interest can be achieved by basing its choice on: (a) a value that may or may not be outside the normal range of occurring values; (b) purely separated graphic information, like drawing elements; (c) separated semantic information, such as a set of spatial and/or temporal coordinates.

Basic Cellular Neural Networks Image Processing

J. Álvaro Fernández
University of Extremadura, Badajoz, Spain

INTRODUCTION

Since its seminal publication in 1988, the Cellular Neural Network (CNN) (Chua & Yang, 1988) paradigm have attracted research community's attention, mainly because of its ability for integrating complex computing processes into compact, real-time programmable analogic VLSI circuits (Rodríguez *et al.*, 2004).

Unlike cellular automata, the CNN model hosts nonlinear processors which, from analogic array inputs, in continuous time, generate analogic array outputs using a simple, repetitive scheme controlled by just a few real-valued parameters. CNN is the core of the revolutionary Analogic Cellular Computer, a programmable system whose structure is the so-called CNN Universal Machine (CNN-UM) (Roska & Chua, 1993). Analogic CNN computers mimic the anatomy and physiology of many sensory and processing organs with the additional capability of data and program storing (Chua & Roska, 2002).

This article reviews the main features of this Artificial Neural Network (ANN) model and focuses on its outstanding and more exploited engineering application: Digital Image Processing (DIP).

BACKGROUND

In the following paragraphs, a definition of the parameters and structure of the CNN is performed in order to clarify the practical usage of the model in DIP.

The standard CNN architecture consists of an $M \times N$ rectangular array of cells $C(i,j)$ with Cartesian coordinates (i,j), $i = 1, 2, \ldots, M, j = 1, 2, \ldots, N$. Each cell or neuron $C(i,j)$ is bounded to a connected neighbourhood or sphere of influence $S_r(i,j)$ of positive integer radius r, which is the set of all neighbouring cells satisfying the following property:

$$S_r(i,j) = \left\{ C(k,l) \;\middle|\; \max_{1 \leq k \leq M, 1 \leq l \leq N} \left\{ |k-i|, |l-j| \right\} \leq r \right\}$$

(1)

This set is sometimes referred as a $(2r+1) \times (2r+1)$ neighbourhood, e.g., for a 3×3 neighbourhood, r should be 1. Thus, the parameter r controls the connectivity of a cell, i.e. the number of active synapses that connects the cell with its immediate neighbours.

When $r > N/2$ and $M = N$, a fully connected CNN is obtained, where every neuron is connected to every other cell in the network and $S_r(i,j)$ is the entire array. This extreme case corresponds to the classic Hopfield ANN model (Chua & Roska, 2002).

The state equation of any cell $C(i,j)$ in the $M \times N$ array structure of the standard CNN may be described mathematically by:

$$C\frac{dz_{ij}(t)}{dt} = -\frac{1}{R}z_{ij}(t) + \sum_{C(k,l) \in S_r(i,j)} \left[A(i,j;k,l) \cdot y_{kl}(t) + B(i,j;k,l) \cdot x_{kl} \right] + I_{ij}$$

(2)

where C and R are values that control the transient response of the neuron circuit (just like an RC filter, typically set to unity for the sake of simplicity), I is generally a constant value that biases or thresholds the state matrix $Z = \{z_{ij}\}$, and S_r is the local neighbourhood of cell $C(i,j)$ defined in (**1**), which controls the influence of the input data $X = \{x_{ij}\}$ and the network output $Y = \{y_{ij}\}$ for time t.

This means that both input and output planes interact with the state of a cell through the definition of a set of real-valued weights, $A(i,j; k, l)$ and $B(i,j; k, l)$, whose size is determined by the neighbourhood radius r. The matrices or cloning templates A and B are called the feedback and feed-forward (or control) operators, respectively.

A standard CNN is typically defined with constant values for r, I, A and B, thus implying that for a fixed input image X, a neuron $C(i, j)$ is provided for each

pixel (i, j), with constant weighted circuits defined by the feedback template A that connects the cell with the output plane Y, and by the control template B, which connects the neuron to the neighbouring pixels of input $x_{ij} \in X$. The value of the neuron state z_{ij} is then adjusted with the bias parameter I, and passed as input to a piecewise-linear function in order to determine the output value y_{ij}. This function may be expressed as

$$y_{ij} = \frac{1}{2}\left(\left|z_{ij}(t)+1\right| - \left|z_{ij}(t)-1\right|\right) \qquad (3)$$

In the Image Processing context, a grey-scale image input X can be represented pixel-wise using a linear map between a pixel value (e.g. a 8-bit integer luminance matrix with 256 grey-scale levels) and the CNN input interval $[-1, +1]$, where the lower limit is used to implement full luminance (i.e. white) and the upper for black pixels (Chua & Yang, 1988).

BASIC CNN IMAGE PROCESSING

The main application of the CNN model, due to its convolution-like scheme, has been DIP modelling and design. In the next subsections a number of basic DIP approaches are introduced, underlining the importance of the network parameters by giving illustrative examples of application. Starting from the standard model described in the previous section, the definition of the standard isotropic CNN follows. Then, an example of application in logic DIP processing is performed in order to introduce the nonlinear effects that implies the using a non-zero feedback template.

The Isotropic CNN Model

For a still image, X will be invariant with time, and for video, $X = X(t)$. In the most general case, r, A, B and I may vary with position and time, and the cloning templates are defined as nonlinear, with the possibility of integrating inhibitory signals for the state matrix and even nonlinear templates that interact with mixed input-output-state data (Chua & Roska, 2002).

These possible extensions raise the definition of a special (and simpler) class of CNN, called isotropic or space-invariant, in which r, A, B and I are fixed for the whole network and where linear synaptic operators are utilized.

In other words,

$$\sum_{C(k,l)\in S_r(i,j)} A(i,j;k,l) \cdot y_{kl} = \sum_{|k-i|\le r} \sum_{|l-j|\le r} A(i-k, j-l) \cdot y_{kl}$$

$$\sum_{C(k,l)\in S_r(i,j)} B(i,j;k,l) \cdot x_{kl} = \sum_{|k-i|\le r} \sum_{|l-j|\le r} B(i-k, j-l) \cdot x_{kl}$$

and $I_{ij} = I$. $\qquad (4)$

The vast majority of the templates defined in the template compendium of (Chua & Roska, 2002) for the CNN-UM are based on this isotropic scheme, using $r = 1$, and binary images in the input plane.

If no feedback (i.e. $A = 0$) is used, then the CNN behaves as a convolution network, using B as a spatial filter, I as a threshold and the piecewise linear output (**3**) as a limiter or saturated output filter. In this way, virtually any spatial filter from DIP theory (Jain, 1989) can be implemented on such a feed-forward driven CNN, which ensures its output stability.

For instance, the EDGE template defined by

$$A = 0, \quad B_{EDGE} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, I = -1 \qquad (5)$$

is designed to work correctly for binary inputs, giving black $(+1)$ output pixels in the input locations where a black edge pixel exists (i.e. if a black pixel has 1 white neighbour), and white (-1) pixels elsewhere.

However, when a grey-scale input image is fed to this CNN, the output may not be a binary image. To solve this potential problem, the following modification is performed over the EDGE CNN:

$$A = 2, B = B_{EDGE}, I = -0.5 \qquad (6)$$

The definition of a centre feedback absolute value greater than 1 in (**6**) ensures a binary output and thus output network stability. The B template used in these CNN is of the *Laplacian* type, having the important property that all surrounding input synaptic weights are inhibitory (i.e. negative) and identical, but the centre synaptic weight is excitatory, and the average of all input synaptic weights is zero.

Apart from edges, convex corners (i.e. black pixels with at least five white neighbours) may also be detected with the following modification of its parameters:

$$A = 2, B = B_{EDGE}, I = -8.5 \qquad (7)$$

This example illustrates the important role played by the threshold parameter *I*. This parameter may be viewed as a bias index that reallocates the origin z_0 of the output function (**3**) (Fernández *et al.*, 2006).

Basic Logic Operators

In order to perform pixel-wise logic operations between two binary images X_1 and X_2, the initial state $Z(0)$ of the network is also utilized as a variable (Chua & Roska, 2002). In standard feed-forward driven CNN, this variable $Z(0)$ is usually set to zero but it can also be used in order to obtain results valid for another applications, such as motion detection and estimation (Torralba & Hérault, 1999).

For example, for a binary set union (logic OR), the following templates are defined:

$$X = X_1, B_1, Z(0) = X_2, A = 3, B = 3, I = 2 (8)$$

whereas for set intersection (logic AND), these variables are defined as

$$X = X_1, Z(0) = X_2, A = 1.5, B = 1.5, I = -1.5 \qquad (9)$$

Once again, the usage of excitatory feedback ensures output stability through the saturation output function (**3**), and the threshold properly biases the final result.

Feedback-Driven Standard CNN

The feedback templates used in all the previously exemplified CNN utilize (if any) only the central element of the template. A standard CNN with off-centre nonzero feedback elements is a special class that exhibits more complex dynamics than those treated so far (Chua & Roska, 1993).

The use of a centre element in *A*, $a_{00} > 1$, means that the output will be binary, i.e. network output will never be stable in the linear region of the saturation function (**3**) (Chua & Roska, 2002). With this restriction, if another element is set in the feedback template,

then two possible situations may occur: the activation of cells in the opposite part of only one of the saturation regions (partial inversion), or wave propagating cell inversions in both binary states.

The first kind of these feedback-driven CNN is said to have the mono-activation property if cells in only one saturated region can enter the linear region. Thus, if cells can enter the linear region from the positive saturation region, then those cells saturated in the negative part must fulfil that the overall contribution of *A*, *B* and *I* in its sphere of influence S_r must be less than –1. That is,

$$w_{ij}(t) = \sum_{S_r(i,j)} \left[a_{kl} \cdot y_{kl}(t) + b_{kl} \cdot x_{kl} \right] + I_{ij} < -1$$

$$(10)$$

On the other hand, if cells enter the linear region only from the negative saturation region, then the contribution for positive stable cells must be $w_{ij}(t) > 1$. It can be demonstrated that in a mono-activated CNN with positive *A* coefficients, with $a_{00} > 1$ and saturated initial values, all the cells that enter the linear region change monotonically their state from (only) one saturated area to the other, and therefore it is a stable nonlinear network (Chua & Roska, 2002).

If, for instance, one element in *A* is negative, the transient will not be monotonic, which does not necessarily imply network instability. An example of a non-monotonic but stable CNN is the Connected Component Detector (CCD) (Matsumoto *et al.*, 1990 a) whose templates (for the horizontal case) are the following:

$$A_{CCD} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}, B = 0, I = 0 \qquad (11)$$

For designing a unidirectional wave propagating mono-activated CNN, a binary activation pattern is defined, which will trigger the transient until output stability is reached (Chua & Roska, 2002). An example of this type of stable feedback-driven CNN is the (horizontal) Shadow Detector (Matsumoto *et al.*, 1990 b) whose parameters are:

$$A_{Shadow} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = 0, I = 0 \qquad (12)$$

FUTURE TRENDS

There is a continuous quest by engineers and specialists: compete with and imitate nature, especially some "smart" animals. Vision is one particular area which computer engineers are interested in. In this context, the so-called Bionic Eye (Werblin *et al.*, 1995) embedded in the CNN-UM architecture is ideal for implementing many spatio-temporal neuromorphic models.

With its powerful image processing toolbox and a compact VLSI implementation (Rodríguez *et al.*, 2004), the CNN-UM can be used to program or mimic different models of retinas and even combinations of them (Lázár *et al.*, 2004). Moreover, it can combine biologically based models, biologically inspired models, and analogic artificial image processing algorithms. This combination will surely bring a broader kind of applications and developments.

CONCLUSION

A number of other advances in the definition and characterization of CNN have been researched in the past decade. This includes the definition of methods for designing and implementing larger than 3×3 neighbourhoods in the CNN-UM (Kék & Zarándy, 1998), the efficient implementation of halftoning techniques (Crounse *et al.*, 1993), the CNN implementation of some image compression techniques (Venetianer *et al.*, 1995) or the design of a CNN-based Fast Fourier Transform algorithm over analogic signals (Perko *et al.*, 1998), between many others. Some of them have also been described in this book in the article entitled *Advanced Cellular Neural Networks Image Processing*.

In this article, a general review of the main properties and features of the Cellular Neural Network model has been addressed, focusing on its DIP capabilities from a basic viewpoint. CNN is now a fundamental and powerful toolkit for real-time nonlinear image processing tasks, mainly due to its versatile programmability, which has powered its hardware development for visual sensing applications.

REFERENCES

Chua, L.O., & Roska, T. (2002). *Cellular Neural Networks and Visual Computing. Foundations and Applications*. Cambridge, UK: Cambridge University Press.

Chua, L.O., & Roska, T. (1993). The CNN Paradigm. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 40, 147–156.

Chua, L.O., & Yang, L. (1988). Cellular Neural Networks: Theory and Applications. *IEEE Transactions on Circuits and Systems*, 35, 1257–1290.

Crounse, K.R., Roska, T., & Chua, L.O. (1993). Image Halftoning with Cellular Neural Networks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* 40, 267-283.

Fernández, J.A., Preciado, V.M., & Jaramillo, M.A. (2006). Nonlinear Mappings with Cellular Neural Networks. *Lecture Notes in Computer Science*, 4177, 350–359.

Jain, A.K. (1989). *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall.

Kék, L., & Zarándy, A. (1998). Implementation of Large Neighborhood Non-Linear Templates on the CNN Universal Machine. *International Journal of Circuit Theory and Applications,* 26, 551-566.

Lázár, A.K., Wagner, R., Bálya, D., & Roska, T. (2004). Functional Representations of Retina Channels via the RefineC Retina Simulator. *International Workshop on Cellular Neural Networks and their Applications CNNA 2004*, 333-338.

Matsumoto, T., Chua, L.O., & Suzuki, H. (1990 a). CNN Cloning Template: Connected Component Detector. *IEEE Transactions on Circuits and Systems,* 37, 633-635.

Matsumoto, T., Chua, L.O., & Suzuki, H. (1990 b). CNN Cloning Template: Shadow Detector. *IEEE Transactions on Circuits and Systems,* 37, 1070-1073.

Perko, M., Iztok Fajfar, I., Tuma, T., & Puhan, J. (1998). Fast Fourier Transform Computation Using a Digital CNN Simulator. *Fifth IEEE International Workshop on Cellular Neural Network and Their Applications Proceedings*, 230-236.

B

Rodríguez, A., Liñán, G., Carranza, L., Roca, E., Carmona, R., Jiménez, F., Domínguez, R., & Espejo, S. (2004). ACE16k: The Third Generation of Mixed-Signal SIMD-CNN ACE Chips Toward VSoCs. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51, 851–863.

Roska, T., & Chua, L.O. (1993). The CNN Universal Machine: An Analogic Array Computer. *IEEE Transactions on Circuits and Systems II: Analog and Digital Processing*, 40, 163–173.

Torralba, A.B., & Hérault, J. (1999). An Efficient Neuromorphic Analog Network for Motion Estimation. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* 46, 269-280.

Venetianer, P.L., Werblin, F., Roska, T., & Chua, L.O. (1995). Analogic CNN Algorithms for Some Image Compression and Restoration Tasks. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* 42, 278-284.

Werblin, F., Roska, T., & Chua, L.O. (1995). The Analogic Cellular Neural Network as a Bionic Eye. *International Journal of Circuit Theory and Applications,* 23, 541-569.

KEY TERMS

Artificial Neural Network (ANN): A system made up of interconnecting artificial neurons or nodes (usually simplified neurons) which may share some properties of biological neural networks. They may either be used to gain an understanding of biological neural networks, or for solving traditional artificial intelligence tasks without necessarily attempting to model a real biological system. Well known examples of ANN are the Hopfield, Kohonen and Cellular (CNN) models.

Feedback: The signal that is looped back to control a system within itself. When the output of the system is fed back as a part of the system input, it is called a feedback loop. A simple electronic device which is based on feedback is the electronic oscillator. The Phase-Locked Loop (PLL) is an example of complex feedback system.

Neuromorphic: A term coined by Carver Mead in the late 1980s to describe VLSI systems containing electronic analogue circuits that mimic neuro-biological architectures present in the nervous system. More recently, its definition has been extended to include both analogue, digital and mixed mode A/D VLSI systems that implements models of neural systems as well as software algorithms.

Piecewise Linear Function: A function $f(x)$ that can be split into a number of linear segments, each of which is defined for a non-overlapping interval of x.

Spatial Convolution: A term used to identify the linear combination of a series of discrete 2D data (a digital image) with a few coefficients or weights. In the Fourier theory, a convolution in space is equivalent to (spatial) frequency filtering.

Template: Also known as kernel, or convolution kernel, is the set of coefficients used to perform a spatial filter operation over a digital image via the spatial convolution operator.

Transient: In electronics, a transient system is a short life oscillation in a system caused by a sudden change of voltage, current, or load. They are mostly found as the result of the operation of switches. The signal produced by the transient process is called the transient signal or simply the transient. Also, the transient of a dynamic system can be viewed as its path to a stable final output.

VLSI: Acronym that stands for Very Large Scale Integration. It is the process of creating integrated circuits by combining thousands (nowadays hundreds of millions) of transistor-based circuits into a single chip. A typical VLSI device is the microprocessor.

Bayesian Neural Networks for Image Restoration

Radu Mutihac
University of Bucharest, Romania

INTRODUCTION

Numerical methods commonly employed to convert experimental data into interpretable images and spectra commonly rely on straightforward transforms, such as the Fourier transform (FT), or quite elaborated emerging classes of transforms, like wavelets (Meyer, 1993; Mallat, 2000), wedgelets (Donoho, 1996), ridgelets (Candes, 1998), and so forth. Yet experimental data are incomplete and noisy due to the limiting constraints of digital data recording and the finite acquisition time. The pitfall of most transforms is that imperfect data are directly transferred into the transform domain along with the signals of interest. The traditional approach to data processing in the transform domain is to ignore any imperfections in data, set to zero any unmeasured data points, and then proceed as if data were perfect.

Contrarily, the maximum entropy (ME) principle needs to proceed from frequency domain to space (time) domain. The ME techniques are used in data analysis mostly to reconstruct positive distributions, such as images and spectra, from blurred, noisy, and/or corrupted data. The ME methods may be developed on axiomatic foundations based on the probability calculus that has a special status as the only internally consistent language of inference (Skilling 1989; Daniell 1994). Within its framework, positive distributions ought to be assigned probabilities derived from their entropy.

Bayesian statistics provides a unifying and self-consistent framework for data modeling. Bayesian modeling deals naturally with uncertainty in data explained by marginalization in predictions of other variables. Data overfitting and poor generalization are alleviated by incorporating the principle of Occam's razor, which controls model complexity and set the preference for simple models (MacKay, 1992). Bayesian inference satisfies the likelihood principle (Berger, 1985) in the sense that inferences depend only on the probabilities assigned to data that were measured and not on the properties of some admissible data that had never been acquired.

Artificial neural networks (ANNs) can be conceptualized as highly flexible multivariate regression and multiclass classification non-linear models. However, over-flexible ANNs may discover non-existent correlations in data. Bayesian decision theory provides means to infer how flexible a model is warranted by data and suppresses the tendency to assess spurious structure in data. Any probabilistic treatment of images depends on the knowledge of the point spread function (PSF) of the imaging equipment, and the assumptions on noise, image statistics, and prior knowledge. Contrarily, the neural approach only requires relevant training examples where true scenes are known, irrespective of our inability or bias to express prior distributions. Trained ANNs are much faster image restoration means, especially in the case of strong implicit priors in the data, nonlinearity, and nonstationarity. The most remarkable work in Bayesian neural modeling was carried out by MacKay (1992, 2003) and Neal (1994, 1996), who theoretically set up the framework of Bayesian learning for adaptive models.

BACKGROUND

Bayesian approach to image restoration is based on the assumption that all of the relevant image information may be stated in probabilistic terms and prior probabilities are known. The ME principle is optimally setting prior probabilities for positive additive distributions. Yet Bayes' theorem and the ME principle share one common future: the updating of a state of knowledge. In some cases, running Bayes' theorem in one hypothesis space and applying the ME principle in another lead to similar calculations.

Neuromorphic and Bayesian modeling may apparently look like extremes of the data modeling spectrum. ANNs are non-linear parallel computational devices endowed with gradient descent algorithms trained by example to solve prediction and classification problems. In contrast, Bayesian statistics is based on coherent

inference and clear axioms. Yet both approaches aim to create models in agreement with data. Bayesian decision theory provides intrinsic means to model ranking. Bayesian inference for ANNs can be implemented numerically by deterministic methods involving Gaussian approximations (MacKay, 1992), or by Monte-Carlo methods (Neal, 1996). Two features distinguish the Bayesian approach to learning models from data. First, beliefs derived from background knowledge are used to select a prior probability distribution for model parameters. Secondly, predictions of future observations are performed by integrating the model's predictions with respect to the posterior parameter distribution obtained by updating this prior with new data. Both aspects are difficult in neural modeling: the prior over network parameters has no obvious relation to prior knowledge, and integration over the posterior is computationally demanding. The properties of priors can be elucidated by defining classes of prior distributions for net parameters that reach sensible limits as the net size goes to infinity (Neal, 1994). The problem of integrating over the posterior can be solved using Markov chain Monte Carlo (Neal 1996).

Bayesian Image Modeling

The fundamental concept of Bayesian analysis is that the plausibility of alternative hypotheses $\{H_i\}_{i \in \mathbb{N}}$ is represented by probabilities $\{P_i\}_{i \in \mathbb{N}}$, and inference is performed by evaluating these probabilities. Inference may opperate on various propositions related in neural modeling to different paradigms. Bayes' theorem makes no reference to any sample or hypothesis space, neither it determines the numerical value of any probability directly from available information. As a prerequisite to apply Bayes' theorem, a principle to cast available information into numerical values is needed.

In statistical restoration of gray-level digital images, the basic assumption is that there exists a scene adequately represented by an orderly array of N pixels. The task is to infer reliable statistical descriptions of images, which are gray-scale digitized pictures and stored as an array of integers representing the intensity of gray level in each pixel. Then the shape of any positive, additive image can be directly identified with a probability distribution. The image is conceived as an outcome of a random vector $f = \{f_1, f_2, ..., f_N\}$, given in the form

of a positive, additive probability density function.

Likewise, the measured data $g = \{g_1, g_2, ... g_M\}$ are expressed in the form of a probability distribution (Fig. 1). Further assumption refers to image data as a linear function of physical intensity, and that the errors (noise) b is data independent, additive, and Gaussian with zero mean and known standard deviation σ_m, $m = 1, 2, ..., M$ in each pixel. The concept of image entropy and the entropy alternative expressions used in image restoration are discussed by Gull and Skilling (1985). A brief review of different approaches based on ME principle, as well as a full Bayesian approach for solving inverse problems are due to Djafari (1995).

Image models are derived on the basis of intuitive ideas and observations of real images, and have to comply with certain criteria of invariance, that is, operations on images should not affect their likelihood. Each model comprises a hypothesis H with some free parameters $w = (\alpha, \beta, ...)$ that assign a probability density $P(f / w, H)$ over the entire image space and normalized to integrate to unity. Prior beliefs about the validity of H before data acquisition are embedded in $P(H)$. Extreme choices for $P(H)$ only may exceed the evidence $P(f / H)$, thus the plausibility $P(H / f)$ of H is given essentially by the evidence $P(f / H)$ of the image f. Consequently, objective means for comparing various hypotheses exist.

Initially, the free parameters w are either unknown or they are assigned very wide prior distributions. The task is to search for the best fit parameter set w_{MP}, which has the largest likelihood given the image. Following Bayes' theorem:

$$P(w / f, H) = \frac{P(f / w, H) \cdot P(w / H)}{P(f / H)} \quad (1)$$

where $P(f / w, H)$ is the likelihood of the image f given w, $P(w / H)$ is the *prior* distribution of w, and $P(f / H)$ is the evidence for H. A prior $P(w / H)$ has to be assigned quite subjectively based on our beliefs about images. Since $P(w / f, H)$ is normalized to 1, then the denominator in (1) ought to satisfy

$$P(f / H) = \int_w P(f / w, H) \cdot P(w / H) \cdot d w .$$ The inte-

Figure 1. Flowchart summarizing the forward and inverse problems
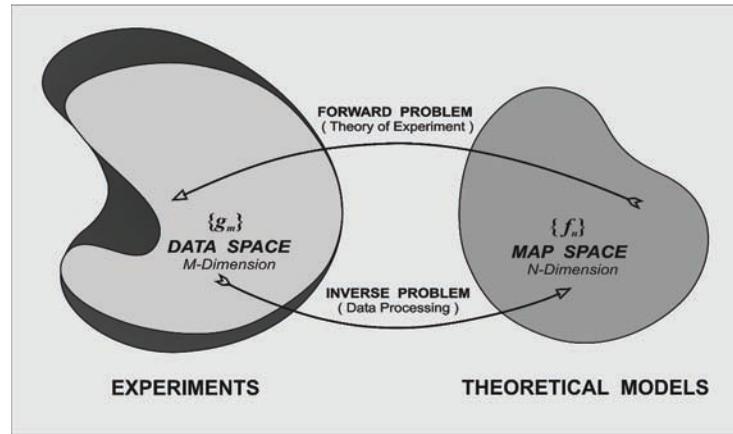
grant is often dominated by the likelihood in w_{MP}, so that the evidence of H is approximated by the best fit likelihood $P(f/w_{MP},H)$ times the Occam's factor (MacKay, 1992):

$$P(f/H) \cong P(f/w_{MP},H) \cdot P(w_{MP}/H) \cdot \Delta w \tag{2}$$

Assuming uniform prior parameter distributions $P(w/H)$ over all admissible parameter sets, then

$$P(w_{MP}) = \frac{1}{\Delta_0 w}, \text{ and the evidence becomes:}$$

$$P(f/H) \cong P(f/w_{MP},H) \cdot \frac{\Delta w}{\Delta_0 w} \tag{3}$$

The ratio

$$\frac{\Delta w}{\Delta_0 w}$$

between the posterior accessible volume of the model's parameter space and the prior accessible volume prevents data overfitting by favoring simpler models. Further, Bayes' theorem gives the probability of H up to a constant:

$$P(H/f) \propto P(f/H) \cdot P(H) \tag{4}$$

Maximum Entropy Methods

Applying the ME principle amounts to assigning a distribution $\{P_1, P_2, ..., P_n\}$ on some hypothesis space by the criterion that it shall maximize some form of entropy subject to constraints that express properties we wish the distribution to have, but are not sufficient to determine it. The ME methods require specifying in advance a definite hypothesis space which sets down the possibilities to be taken into consideration. They come out with a probability distribution, rather than a probability. The ME probability of a single hypothesis H that is not embedded in a space of alternative hypotheses does not make any sense. The ME methods do not require for input the numerical values of any probabilities on that space, rather they assign numerical values to available information as expressed by the choice of hypothesis space and constraints.

LINEAR IMAGING EXPERIMENTS

In the widely spread linear case, where the *N*-dimensional image vector *f* consists of the pixel values of an unobserved image, and the *M*-dimensional data vector *g* is made of the pixel values of an observed image supposed to be a degraded version of *f*, and assuming zero-mean Gaussian additive errors:

$$g = \mathbf{R}f + b \tag{5}$$

where the $M \times N$ matrix $\mathbf{R}$ stands for the PSF (transfer function or instrumetal response) of the imaging system; then the likelihood of data is:

$$P(g/f,\mathbf{C},H) = \frac{1}{(2\pi)^{\frac{M}{2}} \cdot det^{\frac{1}{2}} \mathbf{C}} \cdot exp\left(-\frac{1}{2}(g-f)^T \mathbf{C}^{-1}(g-f)\right) \tag{6}$$

where $\mathbf{C}$ is the covariance matrix of the error vector b. If there is no correlation among the pixels and each pixel has the standard deviation σ_m, $m = 1, 2, ..., M$, then the symmetric full rank covariance matrix becomes diagonal with the elements $C_{mm} = \sigma_m^2$, $m = 1, 2, ..., M$. Hence the probability of the data g given the image f may be written as:

$$P(g/f,\mathbf{C},H) = \frac{1}{(2\pi)^{\frac{M}{2}} \prod_{m=1}^{M} \sigma_m} \cdot exp\left(-\frac{1}{2}\sum_{m=1}^{M} \frac{\left(g_m - \sum_{n=1}^{N} R_{mn}f_n\right)^2}{\sigma_m^2}\right) \tag{7}$$

The full joint posterior $P(f,\theta/g,H)$ of the image f and the unknown PSF parameters denoted generically by θ should be evaluated. Then the required inference about the posterior probability $P(f/g,H)$ is obtained as a marginal integral of this joint posterior over the uncertainties in the PSF:

$$P(f/g,H) = \int P(f,\theta/g,H) \cdot d\theta = \int P(f/\theta,g,H) \cdot P(\theta/g,H) \cdot d\theta \tag{8}$$

Now applying Bayes' theorem for the parameters θ:

$$P(\theta/g,H) = \frac{P(g/\theta,H) \cdot P(\theta/H)}{P(g/H)} \tag{9}$$

and substituting in (8)

$$\int P(f,\theta/g,H) \cdot d\theta \propto \int P(f/\theta,g,H) \cdot P(g/\theta,H) \cdot P(\theta/H) \cdot d\theta \tag{10}$$

If the evidence $P(g/\theta,H)$ is sharply peaked around some value $\hat{\theta}$ and the prior $P(\theta/H)$ is fairly flat in that region, then $P(f/g,H) \cong P(f/\hat{\theta},g,H)$. Otherwise, if the marginal integrant is not well approximated at the modal value of the evidence, then misleading narrow posterior probability densities may result.

If the errors have uniform standard deviation σ_b, then the symmetric covariance matrix has full rank M with $\mathbf{C} = \sigma_b^2\mathbf{I}$, and the probability of data (7) becomes:

$$P(g/f,\beta,H) = \frac{1}{Z_b(\beta)} \cdot exp\left(-\sum_{m=1}^{M} \beta \cdot E_b(g/f,H)\right) \tag{11}$$

where $\beta = 1/\sigma_b^2$ is a measure of the noise in each pixel,

$$E_b(g/f,H) = \frac{1}{2} \cdot \frac{b^T b}{\sigma_b^2} = \frac{1}{2}\sum_{m=1}^{M} \frac{\left(g_m - \sum_{n=1}^{N} R_{mn}f_n\right)^2}{\sigma_b^2}$$

is the error function, and is the noise partition function.

More complex models use the intrinsic correlation function $\mathbf{C} = \left[\mathbf{GG}^T\right]^{-1}$, where $\mathbf{G}$ is a convolution from an imaginary hidden image, which is uncorrelated, to the real correlated image.

If the prior probability of the image f is also Gaussian:

$$P(f/\mathbf{F}_0,H) = \frac{1}{(2\pi)^{\frac{N}{2}} \cdot det^{\frac{1}{2}} \mathbf{F}_0} \cdot exp\left(-\frac{1}{2}f^T \mathbf{F}_0^{-1} f\right) \tag{12}$$

where is the prior covariance matrix of f, and assuming a uniform standard deviation of the image, then its prior probability distribution becomes:

$$P(f/\alpha,H) = \frac{1}{Z_f(\alpha)} \cdot exp\left(-\alpha E_f(f/\mathbf{F}_0)\right) \tag{13}$$

where the parameter $\alpha = 1/\sigma_f^2$ measures the expected smoothness of f, $Z_f(\alpha) = (2\pi/\alpha)^{N/2}$ is the partition

B

function of f, and

$$E_f\left(f/\mathbf{F}_0\right)=\frac{1}{2}f^T\,\mathbf{F}_0^{-1}\,f\;.$$

The posterior probability of image f given data g is derived from Bayes' theorem:

$$P\left(f/g,\alpha,\,\beta,H\right)=\frac{P\left(g/f,\beta,H\right)\cdot P\left(f/\alpha,H\right)}{P\left(g/\alpha,\,\beta,H\right)}$$

(14)

where the evidence $P\left(g/\alpha,\,\beta,H\right)$ is the normalizing factor. Since the denominator in (14) is a product of Gaussian functions of f, we may rewrite:

$$P\left(f/g,\alpha,\,\beta,H\right)=\frac{exp\,\left(-\alpha E_f-\beta E_b\right)}{Z_M\left(\alpha,\,\beta\right)}=\frac{exp\,\left(-M\left(f\right)\right)}{Z_M\left(\alpha,\,\beta\right)}$$

(15)

where

$$M\left(f\right)=\alpha\,E_f+\beta E_b$$
$$\text{and } Z_M\left(\alpha,\,\beta\right)=\int_f exp\,\left(-M\left(f\right)\right)\cdot d\,f$$

with the integral covering the space of all admissible images in the partition function. Therefore, minimizing the objective function $M(f)$ corresponds to finding the most probable image f_{MP}, which is the mean value of the Gaussian posterior distribution. Its covariance matrix $\mathbf{A}^{-1}$ that defines the joint error bars on f can be obtained from the Hessian matrix $\mathbf{A}=-\nabla\nabla log\,P\left(f/g,\alpha,\,\beta,H\right)$ evaluated at f_{MP}. The image f_{MP} is obtained by differentiating $log\,P\left(f/g,\alpha,\,\beta,H\right)$ and solving for the derivative being zero:

$$f_{MP}=\left[\mathbf{R}^T\mathbf{R}-\frac{\sigma_b^2}{\sigma_f^2}\mathbf{C}\right]^{-1}\mathbf{R}^T\,f$$

(16)

The term

$$\frac{\sigma_b^2}{\sigma_f^2}\mathbf{C}$$

regularizes the ill-conditioned inversability. When the term

$$\frac{\sigma_b^2}{\sigma_f^2}\mathbf{C}$$

is negligible, the optimal linear filter

$$\left[\mathbf{R}^T\mathbf{R}-\frac{\sigma_b^2}{\sigma_f^2}\mathbf{C}\right]^{-1}\mathbf{R}^T$$

equates to the pseudoinverse $\mathbf{R}^{-1}=\left[\mathbf{R}^T\mathbf{R}\right]^{-1}\mathbf{R}^T$.

Entropic Prior of Images

Invoking the ME principle requires that the prior knowledge to be stated as a set of constraints on f, though affecting the amount by which the image reconstruction is offset from reality. The prior information about f may be expressed as a probability distribution (Djafari, 1995):

$$P\left(f/\alpha,H\right)=\frac{1}{Z\left(\alpha\right)}\cdot exp\,\left(-\alpha\cdot\Phi\left(f\right)\right)\quad(17)$$

where α is generally a positive parameter and $Z(\alpha)$ is the normalizing factor. The entropic prior in the discrete case may correspond to potential functions like:

$$\Phi(f)=\sum_{n=1}^{N}f_n\cdot ln\,\frac{f_n}{U}$$

(18)

where U is the total number of quanta in the image f (Mutihac *et al.*, 1997).

The posterior probability of an image f drawn from some measured data g is given by Bayes' theorem:

$$P\left(f/g,\alpha,\mathbf{C},H\right)\propto exp\,\left(-\alpha\sum_{n=1}^{N}f_n\cdot ln\,\left(\frac{f_n}{U}\right)\right)\cdot exp\,\left(-\frac{1}{2}\sum_{m=1}^{M}\frac{\left(g_m-\sum_{n=1}^{N}R_{mn}f_n\right)}{\sigma_m^2}\right)$$

(19)

An estimation rule, such as posterior mean or maximum a posteriori (MAP), is needed in order to choose an optimal, unique, and stable solution $\tilde{f}$ for the estimated image. The posterior probability is assumed to summarize the full state of knowledge on a given scene. Producing a single image as the best restoration naturally leads to the most likely one which maximizes the posterior probability $P\left(f/g,\alpha,\mathbf{C},H\right)$,

along with some statement of reliability derived from the spread of all admissible images.

In variational problems with linear constraints, Agmon *et al*. (1979) showed that the potential function associated to a positive, additive image is always concave for any set of Lagrange multipliers, and it possesses an unique minimum which coincides with the solution of the nonlinear system of constraints. As a prerequisite, the linear independence of the constraints is checked and then the necessary and sufficient conditions for a feasible solution are formulated. Wilczek and Drapatz (1985) suggested the Newton-Raphson's iteration method as offering high accuracy results. Ortega and Rheinboldt (1970) adopted a continuation technique for the very few cases where the Newton's method fails to converge. These techniques are nevertheless successful in practice for relatively small data sets only and assume a symmetric positive definite Hessian matrix of the potential function.

Quality Assessment of Image Restoration

In all digital imaging systems, quality degradation is inevitably due to various sources like photon shot noise, finite acquisition time, readout noise, dark current noise, and quantization noise. Some noise sources can be effectively suppressed yet some cannot. The combined effect of these degradation sources is often modeled by Gaussian additive noise (Pham *et al*. 2005).

In order to quantitatively estimate the restoration quality in the case of similar size ($M = N$) for both the measured g and the restored image $\tilde{f}$, the mean energy of restoration error:

$$D = \frac{1}{N} \sum_{n=1}^{N} \left[g_n - \tilde{f}_n \right]^2 \qquad (20)$$

may be used as a merit factor. Yet too high a value for D may set the restored image quite away from the original scene and raise questions on introducing spurious features for which there is no clear evidence in measurements and complicating the subsequent inference and plausibility.

A more realistic degradation measure of image blurring by additive noise is referred to in terms of a metric

called blurred signal-to-noise ratio redefined here by using the noise variance in each pixel such as:

$$BSNR = 10 \cdot lg \; \frac{1}{N} \sum_{n=1}^{N} \frac{\left[y_n - \bar{y}_n \right]^2}{\sigma_n^2} \qquad (21)$$

where $y = g - b$ is the difference between the measured data g and the noise b.

In simulations, where the original image f of the measured data g is available, the objectivity of testing the performance of image restoration algorithms may be assessed by the improvement of signal-to-noise ratio metric defined as:

$$ISNR = 10 \cdot lg \; \frac{\sum_{n=1}^{N} \left[f_n - g_n \right]^2}{\sum_{n=1}^{N} \left[f_n - \tilde{f}_n \right]^2} \qquad (22)$$

where $\tilde{f}$ is the best statistical estimation of the correct solution f.

While mean squared error metrics like ISNR do not always reflect the perceptual properties of the human visual system, they may provide an objective standard by which to compare different image processing techniques. Nevertheless, it is of major significance that various algorithms behavior be analyzed from the point of view of ringing and noise amplification, which can be a key indicator of improvement in quality for subjective comparisons of restoration algorithms (Banham and Katsaggelos, 1997).

FUTURE TRENDS

A practical Bayesian framework for neural-inspired modeling aims to develop probabilistic models that fit data and perform optimal predictions. The link between Bayesian inference and neural models gives new perspectives to the assumptions and approximations made on ANNs when used as associative memories. Evolutionary optimization algorithms capable to discover absolute function minimum (maximum) are needed.

A statistically biased redefinition of the concept of pattern existence used in a quantitative manner to assess the overall quality of digital images with domain-specific relevance would increase the accuracy of ranking the image restoration methods.

An efficient MAP procedure has to be implemented in a recursive supervised trained neural net to get restored (reconstructed) the best image in compliance with the existing constraints, measuring and modeling errors.

CONCLUSION

A major intrinsic difficulty in Bayesian image restoration resides in determination of a prior law for images. The ME principle solves this problem in a self-consistent way. The ME model for image deconvolution enforces the restored image to be positive. The spurious negative areas and complementary spurious positive areas are wiped off and the dynamic range of the restored image is substantially enhanced.

Image restoration based on image entropy is effective even in the presence of significant noise, missing or corrupted data. This is due to the appropriate regularization of the inverse problem of image restoration introduced in a coherent way by the ME principle. It satisfies all consistency requirements when combining the prior knowledge and the information contained in experimental data. A major result is that no artifacts are added since no structure is enforced by entropic priors.

Bayesian ME approach is a statistical method which directly operates in spatial domain, thus eliminating the inherent errors coming out from numerical Fourier direct and inverse transformations and from the truncation of signals.

REFERENCES

Agmon, N., Alhassid, Y., & Levine, R. D. (1979). An algorithm for finding the distribution of maximal entropy. *Journal of Computational Physics, 30*, 250-258.

Banham, M. R. & Katsaggelos, A. K. (1997, March). Digital image restoration, *IEEE Signal Processing Magazine*, 24-41.

Berger, J. (1985). *Statistical decision theory and Bayesian analysis*. Springer-Verlag.

Candes, E. J. (1993). Ridgelets: Theory and applications. *PhD Thesis*. Department of Statistics, Standford University, 1998.

Daniell, G. J. (1994). Of maps and monkeys: An introduction to the maximum entropy method. In B. Buck & V. A. Macaulay (Eds.), *Maximum entropy in action* (pp. 1-18). Oxford: Clarendon Press.

Djafari, A. M.- (1995). A full Bayesian approach for inverse problems. In K. M. Hanson & R. N. Silver (Eds.), *Maximum entropy and bayesian methods* (pp. 135-144).

Donoho, D. L. (1996). Unconditional bases and bit-level compression. *Applied and Computational Harmonic Analysis*, *1*(1), 100-105.

Gull, S. F. & Skilling, J. (1985). The entropy of an image. In C. R. Smith & W. T. Grandy Jr. (Eds), *Maximum entropy and Bayesian methods in inverse problems* (pp. 287-302), Dordrecht: Kluwer Academic Publishers.

MacKay, D. J. K. (1992). A practical Bayesian framework for backpropagation networks, *Neural Computation*, *4*, 448-472.

MacKay, D. J. K. (2003). *Information theory, inference, and learning algorithms*. Cambridge: University Press.

Mallat, S. (2000). *Une exploration des signaux en ondelettes*, Editions de l'Ecole Polytechnique.

Mayers, K. J. & Hanson, K. M. (1990). Comparison of the algebraic reconstruction technique with the maximum entropy reconstruction technique for a variety of detection tasks. *Proceedings of SPIE*, *1231*, 176-187.

Meyer, Y. (1993). Review of "An introduction to wavelets and ten lectures on wavelets." *Bulletin of the American Mathematical Society*, *28*, 350-359.

Mutihac, R., Colavita A.A., Cicuttin, A. & Cerdeira, A. E. (1997). Bayesian modeling of feed-forward neural networks. *Fuzzy Systems & Artificial Intelligence*, *6*(1-3), 31-40.

Neal, R.M. (1994). Priors for infinite networks. *Technical Report* CRG-TR-94-1, Department of Computer Science, University of Toronto.

Neal, R. M. (1996). Bayesian learning for neural networks. In *Lecture Notes in Statistics*, 118, New York: Springer-Verlag

Ortega, J. M. & Rheinboldt, W. B. (1970). *Iterative solution of nonlinear equations in several variables*. New York: Academic Press.

Pham, T. Q., van Vliet, L. J., & Schutte K. (2005). Influence of SNR and PSF on limits of super-resolution. *Proceedings of SPIE-IS&T Electronic Imaging*, 5672, 169-180.

Skilling, J. (1989). Classic maximum entropy. In J. Skilling (Ed.), *Maximum entropy and Bayesian methods* (pp. 45-52), Dordrecht: Kluwer Academic Publishers.

Wilczek, R. & Drapatz, S. (1985). A high accuracy algorithm for maximum entropy image restoration in the case of small data sets. *Astronomy and Astrophysics*, *142*, 9-12.

KEY TERMS

Artificial Neural Networks (ANNs): Highly parallel nets of interconnected simple computational elements, which perform elementary operations like summing the incoming inputs (afferent signals) and amplifying/thresholding the sum.

Bayesian Inference: An approach to statistics in which all forms of uncertainty are expressed in terms of probability.

Deconvolution: An algorithmic method for eliminating noise and improving the resolution of digital data by reversing the effects of convolution on recorded data.

Digital Image: A representation of a 2D/3D image as a finite set of digital values called pixels/voxels typically stored in computer memory as a raster image or raster map.

Entropy: A measure of the uncertainty associated with a random variable. Entropy quantifies information in a piece of data.

Image Restoration: A blurred image can be significantly improved by deconvolving its PSF in such a way that the result is a sharper and more detailed image.

Point Spread Function (PSF): The output of the imaging system for an input point source.

Probabilistic Inference: An effective approach to approximate reasoning and empirical learning in AI.

Behaviour–Based Clustering of Neural Networks

María José Castro-Bleda
Universidad Politécnica de Valencia, Spain

Slavador España-Boquera
Universidad Politécnica de Valencia, Spain

Francisco Zamora-Martínez
Universidad Politécnica de Valencia, Spain

INTRODUCTION

The field of off-line optical character recognition (OCR) has been a topic of intensive research for many years (Bozinovic, 1989; Bunke, 2003; Plamondon, 2000; Toselli, 2004). One of the first steps in the classical architecture of a text recognizer is preprocessing, where noise reduction and normalization take place. Many systems do not require a binarization step, so the images are maintained in gray-level quality. Document enhancement not only influences the overall performance of OCR systems, but it can also significantly improve document readability for human readers. In many cases, the noise of document images is heterogeneous, and a technique fitted for one type of noise may not be valid for the overall set of documents. One possible solution to this problem is to use several filters or techniques and to provide a classifier to select the appropriate one.

Neural networks have been used for document enhancement (see (Egmont-Petersen, 2002) for a review of image processing with neural networks). One advantage of neural network filters for image enhancement and denoising is that a different neural filter can be automatically trained for each type of noise.

This work proposes the clustering of neural network filters to avoid having to label training data and to reduce the number of filters needed by the enhancement system. An agglomerative hierarchical clustering algorithm of supervised classifiers is proposed to do this. The technique has been applied to filter out the background noise from an office (coffee stains and footprints on documents, folded sheets with degraded printed text, etc.).

BACKGROUND

Multilayer Perceptrons (MLPs) have been used in previous works for image restoration: the input to the MLP is the pixels in a moving window, and the output is the restored value of the current pixel (Egmont-Petersen, 2000; Hidalgo, 2005; Stubberud, 1995; Suzuki, 2003). We have also used neural network filters to estimate the gray level of one pixel at a time (Hidalgo, 2005): the input to the MLP consisted of a square of pixels that was centered at the pixel to be cleaned, and there were four output units to gain resolution (see Figure 1). Given a set of noisy images and their corresponding clean counterparts, a neural network was trained. With the trained network, the entire image was cleaned by scanning all the pixels with the MLP. The MLP, therefore, functions like a nonlinear convolution kernel. The universal approximation property of a MLP guarantees the capability of the neural network to approximate any continuous mapping (Bishop, 1996).

This approach clearly outperforms other classic spatial filters for reducing or eliminating noise from images (the mean filter, the median filter, and the closing/opening filter (Gonzalez, 1993)) when applied to enhance and clean a homogeneous background noise (Hidalgo, 2005).

BEHAVIOUR-BASED CLUSTERING OF NEURAL NETWORKS

Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering is considered to be a more convenient approach than other clustering

Figure 1. An example of document enhancement with an artificial neural network. A cleaned image (right) is obtained by scanning the entire noisy image (left) with the neural network.
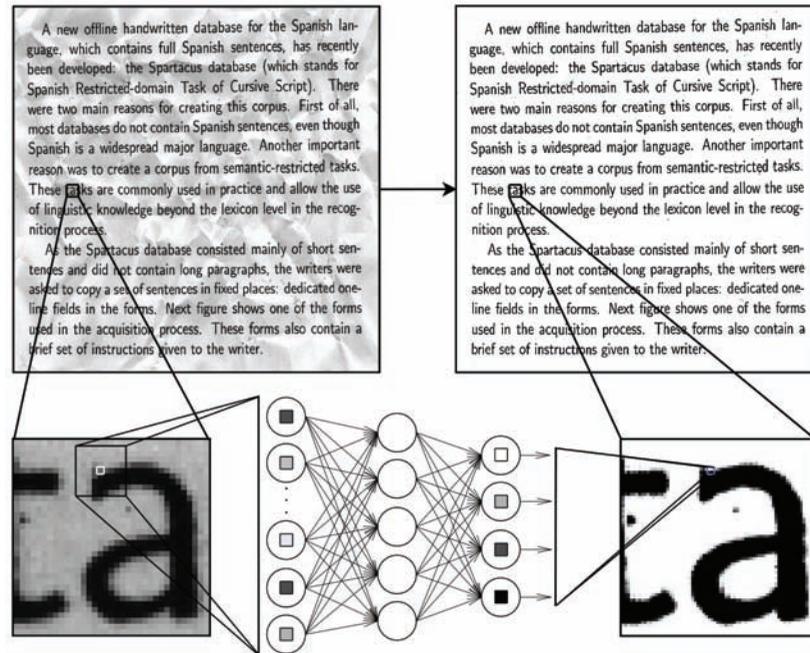
algorithms, mainly because it makes very few assumptions about the data (Jain, 1999; Mollineda, 2000). Instead of looking for a single partition (based on finding a local minimum), this clustering algorithm constructs a hierarchical structure by iteratively merging clusters according to certain dissimilarity measure, starting from singletons until no further merging is possible (one general cluster). The hierarchical clustering process can be illustrated with a tree that is called dendogram, which shows how the samples are merged and the degree of dissimilarity of each union (see Figure 2). The dendogram can be easily broken at a given level to obtain clusters of the desired cardinality or with a specific dissimilarity measure. A general hierarchical clustering algorithm can be informally described as follows:

1. Initialization: *M* singletons as *M* clusters.
2. Compute the dissimilarity distances between every pair of clusters.
3. Iterative process:

a) Determine the closest pair of clusters *i* and *j*.
b) Merge the two closest clusters into a new cluster *i+j*.
c) Update the dissimilarity distances from the new cluster *i+j* to all the other clusters.
d) If more than one cluster remains, go to step a).

4. Select the number *N* of clusters for a given criterion.

Behaviour-Based Clustering of Supervised Classifiers

When the points of the set to be clustered are supervised classifiers, both a dissimilarity distance and the way to merge two classifiers must be defined (see Figure 2):

1. The dissimilarity distance between two clusters can be based on the behaviour of the classifiers with respect to a validation dataset. The more similar the output of two classifiers is, the closer they are.
2. To merge the closest pair of clusters, a new classifier is trained with the associated training data

Figure 2. Behaviour-based clustering of supervised classifiers. An example of the dendogram obtained for M=5 points: A, B, C, D, E. If N=3, three clusters are selected: A+B, C, D+E. In this work, to merge two clusters, a new classifiers is trained. For example, cluster D+E is trained with the data used to train the classifiers D and E.
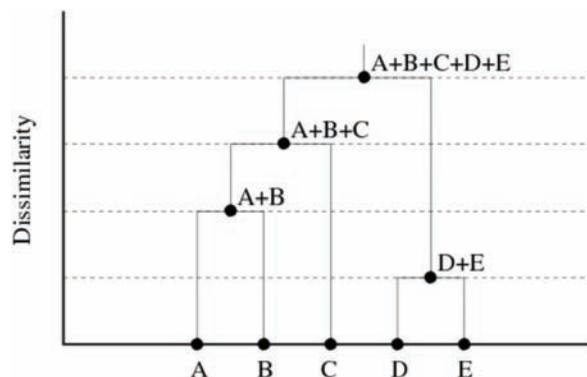
of both clusters. Another possibility is to build an ensemble of the two classifiers.

An Application of Behaviour-based Clustering of MLPs to Document Enhancement

In this work, MLPs are used as supervised classifiers. When two clusters are merged, a new MLP is trained with the associated training data of the two merged MLPs.

This behaviour-based clustering algorithm has been applied to enhance printed documents with typical noises from an office (folded sheets, wrinkled sheets, coffee stains, ...). Figure 1 shows an example of a noisy printed document (wrinkled sheet) from the corpus.

A set of MLPs is trained as neural filters for different types of noise and then clustered into groups to obtain a reduced set of neural clustered filters. In order to automatically determine which clustered filter is the most suitable to clean and enhance a real noisy image, an image classifier is also trained using MLPs. Experimental results using this enhancement system show excellent results in cleaning noisy documents (Zamora-Martínez, 2007).

FUTURE TRENDS

Document enhancement is becoming more and more relevant due to the huge amount of scanned documents. Besides, it not only influences the overall performance of **OCR** systems, but it can also significantly improve document readability for human readers.

The method proposed in this work can be improved twofold: by using ensembles of MLPs when two MLPs are merged, and by improving the method to select the neural clustered filter that is the most suitable to enhance a given noisy image.

CONCLUSION

An agglomerative hierarchical clustering of supervised-learning classifiers that uses a measure of similarity among classifiers based on their behaviour on a validation dataset has been proposed. As an application of this clustering procedure, we have designed an enhancement system for document images using neural network filters. Both objective and subjective evaluations of the cleaning method show excellent results in cleaning noisy documents. This method could also be used to clean and restore other types of images, such as noisy backgrounds in scanned documents, stained paper of historical documents, vehicle license recognition, etc.

REFERENCES

Bishop, C.M. (1996). Neural Networks for Pattern Recognition. Oxford University Press.

Bozinovic, R.M., & Srihari, S.N. (1989). Off-Line Cursive Script Word Recognition. IEEE Trans. on PAMI, 11(1), 68–83.

Bunke, H. (2003). Recognition of Cursive Roman Handwriting – Past, Present and Future. In: Proc. ICDAR. 448–461.

Egmont-Petersen, M., de Ridder, D., & Handels, H. (2002). Image processing with neural networks – a review. Pattern Recognition 35(10). 2279–2301.

Gonzalez, R., & Woods, R. (1993). Digital Image Processing. Addison-Wesley Pub. Co.

Hidalgo, J.L., España, S., Castro, M.J., & Pérez, J.A. (2005). Enhancement and cleaning of handwritten data by using neural networks. In: Pattern Recognition and Image Analysis. Volume 3522 of LNCS. Springer-Verlag. 376–383

Jain, A.K., Murty, M.N., & Flynn, P.J. (1999). Data clustering: a review. ACM Comput. Surv. 31(3). 264–323

Kanungo, T., & Zheng, Q. (2004). Estimating Degradation Model Parameters Using Neighborhood Pattern Distributions: An Optimization Approach. IEEE Trans. on PAMI 26(4). 520–524.

Mollineda, R.A., & Vidal, E. (2000). A relative approach to hierarchical clustering. In: Pattern Recognition and Applications. Volume 56. IOS Press. 19–28.

Plamondon, R., & Srihari, S.N. (2000). On-line and off-line handwriting recognition: A comprehensive survey. IEEE Trans. on PAMI 22(1). 63–84.

Stubberud, P., Kanai, J., & Kalluri, V. (1995). Adaptive Image Restoration of Text Images that Contain Touching or Broken Characters. In: Proc. ICDAR. Volume 2. 778–781.

Suzuki, K., Horiba, I., & Sugie, N. (2003). Neural Edge Enhancer for Supervised Edge Enhancement from Noisy Images. IEEE Trans. on PAMI 25(12). 1582–1596.

Toselli, A.H., Juan, A., González, J., Salvador, I., Vidal, E., Casacuberta, F., Keysers, D., & Ney, H. (2004). Integrated Handwriting Recognition and Interpretation using Finite-State Models. Int. Journal of Pattern Recognition and Artificial Intelligence 18(4). 519-539.

F. Zamora-Martínez, S. España-Boquera, & M.J. Castro-Bleda. (2007). Behaviour-based Clustering of Neural Networks applied to Document Enhancement. In: Computational and Ambient Intelligence. Volume 4507 of LNCS. Springer-Verlag. 144-151.

http://en.wikipedia.org

KEY TERMS

Artificial Neural Network: An artificial neural network (ANN), often just called a "neural network" (NN), is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing based on a connectionist approach to computation.

Backpropagation Algorithm: A supervised learning technique used for training artificial neural networks. It was first described by Paul Werbos in 1974, and further developed by David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams in 1986. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop).

Clustering: The classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

Document Enhancement: Accentuation of certain desired features, which may facilitate later processing steps such as segmentation or object recognition.

Hierarchical Agglomerative Clustering: Hierarchical Clustering algorithms find successive clusters using previously established clusters. Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.

Multilayer Perceptron (MLP): This class of artificial neural networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function.

B

Optical Character Recognition (OCR): A type of computer software designed to translate images of handwritten or typewritten text (usually captured by a scanner) into machine-editable text, or to translate pictures of characters into a standard encoding scheme representing them (e.g. ASCII or Unicode). OCR began as a field of research in pattern recognition, artificial intelligence and machine vision.

Supervised Learning: A machine learning technique for creating a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs. The output of the function can be a continuous value (called regression), or can predict a class label of the input object (called classification). The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output).

Bio-Inspired Algorithms in Bioinformatics I

José Antonio Seoane Fernández
University of A Coruña, Spain

Mónica Miguélez Rico
University of A Coruña, Spain

INTRODUCTION

Large worldwide projects like the Human Genome Project, which in 2003 successfully concluded the sequencing of the human genome, and the recently terminated Hapmap Project, have opened new perspectives in the study of complex multigene illnesses: they have provided us with new information to tackle the complex mechanisms and relationships between genes and environmental factors that generate complex illnesses (Lopez, 2004; Dominguez, 2006).

Thanks to these new genomic and proteomic data, it becomes increasingly possible to develop new medicines and therapies, establish early diagnoses, and even discover new solutions for old problems. These tasks however inevitably require the analysis, filtration, and comparison of a large amount of data generated in a laboratory with an enormous amount of data stored in public databases, such as the NCBI and the EBI.

Computer sciences equip biomedicine with an environment that simplifies our understanding of the biological processes that take place in each and every organizational level of live matter (molecular level, genetic level, cell, tissue, organ, individual, and population) and the intrinsic relationships between them.

Bioinformatics can be described as the application of computational methods to biological discoveries (Baldi, 1998). It is a multidisciplinary area that includes computer sciences, biology, chemistry, mathematics, and statistics. The three main tasks of bioinformatics are the following: develop algorithms and mathematical models to test the relationships between the members of large biological datasets, analyze and interpret heterogeneous data types, and implement tools that allow the storage, retrieve, and management of large amounts of biological data.

BACKGROUND

The following section describes some of the problems that are most commonly found in bioinformatics.

Interpretation of Gene Expression

The expression of genes is the process by which the codified information of a gene is transformed into the necessary proteins for the development and functioning of the cell. In the course of this process, small sequences of ARN, also called ARN messengers, are formed by transcription and subsequently *translated* into proteins.

The amount of expressed mARN can be measured with various methods, such as gel electrophoresis, but large numbers of simultaneous expression analyses are usually carried out with microarrays (Quackenbush, 2001), which make it possible to obtain the simultaneous expression of tens of thousands of genes; such an amount of data can only be analyzed with the help of an informatic process.

Among the most common tasks in this type of analysis is the task to find the differences between, for instance, a patient and a test that determines whether a gene is expressed or not. These tasks can be divided into classical problems of classification and clustering. Clustering is used not only in experiments of microarrays (to identify groups of genes with similar expressions), but also suggests functional relationships between the members of the cluster.

Alignment of ADN, ARN, and Protein Sequences

Sequences alignment consists in superposing two or more sequences of both nucleotides (ADN and ARN) and amino acids (proteins) in order to compare them and analyze the sequence parts that are alike and unalike.

The optimal alignment is that which mainly shows correspondences between the nucleotides or amino acids and is therefore said to have the highest score. This alignment may or may not have a biological meaning. There are two types of alignment: the global alignment, which maximizes the number of coincidences in the entire sequence, and the local alignment, which looks for similar regions in large sequences that are normally highly divergent. The most commonly used technique to implement alignments is dynamic programming by means of the Smith-Waterman algorithm (Smith, 1981), which explores all the possible comparisons in the sequences.

Another problem in sequences alignment is multiple alignment (Wallace, 2005), which consists in aligning three or more sequences of ADN, ARN, or proteins, and is generally used to search for evolutive relationships between these sequences. The problem is equivalent to that of simple sequences alignment, but takes into consideration the n sequences that are to be compared. The complexity of the algorithm increases exponentially with the number of sequences to compare.

Identification of the Gene Regulatory Network

All the information of a living organism's genome is stored in each and every one of its cells. Whereas the genome is used to synthesize information on all the body cells, the regulating network is in charge of guiding the expression of a given set of genes in one cell rather than another so as to form certain types of cells (cellular differentiation) or carry out specific functions related to spatial and temporal localization; in other words, it makes the genes express themselves when and where necessary. The role of a gene regulatory network therefore consists in integrating the dynamic behaviour of the cell and the external signals with the environment of the cell, and to guide the interaction of all the cells so as to control the process of cellular differentiation (Geard, 2004). Inferring this regulating network from the cellular expression data is considered to be one of the most complex problems in bioinformatics (Akustsu, 1999).

Construction of Phylogenetic Trees

A phylogenetic tree (Setúbal, 1999) is a tree that shows the evolutionary relationships between various spe-

cies of individuals that are believed to have common descendence. Whereas traditionally morphological characteristics are used to carry out such analyses, in the present case we will study molecular phylogenetic trees, which use sequences of nucleotides or amino acids for classification. The construction of these trees is initially based on algorithms for multiple sequences alignment, which allows us to classify the evolutive relationships between homologue genes present in various species. In a second phase, we must calculate the genetic distance between each pair of sequences in order to represent them correctly in the tree.

Gene Finding and Mapping

Gene finding (Fickett, 1996) basically consists in identifying genes in an ADN chain by recognizing the sequence that initiates the codification of the gene or *gene promoter*. When the protein that will interpret the gene finds the sequence of that promoter, we know that the next step is the recognition of the gene.

Gene mapping (Setúbal, 1999) consists in creating a genetic map by assigning genes to a position inside the chromosome and by indicating the relative distance between them. There are two types of mapping. Physical or *cytogenetic* mapping, on the one hand, consists in dividing the chromosome into small labelled fragments. Once divided, they must be ordered and situated in their correct position in the chromosome. Link mapping, on the other hand, shows the position of some genes with respect to others. The latter mapping type has two inconveniences: it does not provide the distance between the genes, and it is unable to provide the correct order if the genes are very close to each other.

Prediction of DNA, RNA, and Protein Structure

The DNA and RNA sequences are folded into a tridimensional structure that is determined by the order of the nucleotides within the sequence. Under the same environmental conditions, the tridimensional structure of these sequences implies a diverging behaviour. Since the secondary structure of the nucleic acids is a factor that affects the link of both DNA molecules and RNA molecules, it is essential to know these structures in order to analyze a sequence.

The prediction of the folds that determine the RNA structure is an important factor in the understanding of

many biological processes, such as translation in the RNA Messenger, replication of RNA chains in viruses, and the function of structural RNA and RNA/proteins complexes.

The tridimensional structure of proteins is extremely diverse, going from completely fibrous to nodular. Predicting the folds of proteins is important, because a protein's structure is closely related to its function. The experimental determination of the proteinic structure as such helps us to find the proteinic function and allows us to design synthetic proteins that can be used as medicines.

BIO-INSPIRED ALGORITHMS

The basic principle of bio-inspired algorithms is to use analogies with natural systems in order to solve problems. By simulating the behaviour of natural systems, these algorithms design heuristic, non-deterministic methods for searching, learning, behaviour, etc. (Forbes, 2004).

Artificial Neural Networks

Artificial neural networks (McCulloch, 1943)(Hertz, 1991)(Bishop, 1995) (Rumelhart, 1986) (ANNs) are computational models inspired on the behaviour of the nervous system. Even though their development is based on the modelling of biological processes in the brain, there are considerable differences between the processing elements of ANNs and actual neurons.

ANNs consist of unit networks that are interconnected and organized in layers that evolve in the course of time. The main features of these systems are the following:Self-Organization and Adaptability: Allow robust and adaptive processing, adaptive training, and self-organizing networks; Non-linear processing: Increase the network's capacity to approach, classify, and be immune to noise;Parallel processing: use a large number of processing units with a high level of interconnectivity.

ANNs can be classified according to their learning type: Supervised learning neural networks: the network learns relationships between the input and output data. The input data are passed on to the input layer and propagate through the network architecture until they reach the output layer. The output obtained in this output layer is compared to the expected output,

and subsequently the weights of the interconnections are modified so as to minimize the error between the obtained and the expected output; Non-supervised learning networks: In this type of learning, none of the expected output types is passed on to the network, but the network itself searches for the differences between the inputs and separates the data accordingly.

Evolutionary Computation

Evolutionary computation (Rechenberg, 1971)(Holland, 1975) is a technique that is inspired on evolutive biological strategies: genetic algorithms, for example, use biological techniques of cross-over, mutation, and selection to solve searching and optimization problems. Each of these operators has an impact on one or more chromosomes, i.e. possible solutions to the problem, and generates another series of chromosomes, i.e. the following generation of solutions. The algorithm is executed iteratively and as such takes the population through the generations until it finds an optimal solution. Another strategy of evolutionary computation is genetic programming (Koza 1990), which uses the same operators as the genetic algorithms to develop the optimal program to solve a problem.

Swarm Intelligence

Swarm intelligence (Beni, 1989)(Bonabeau, 2001)(Engelbrench, 2005) is a recent family of bio-inspired techniques based on the social or collective behaviour of groups such as ants, bees, etc., insects which have very limited capacities as individuals, but form groups to carry out complex tasks.

Immune Artificial System

The immune artificial system (Farmer, 1986)(Dasgupta, 1999) is a new computational paradigm that has appeared in recent years and is based on the immune system of vertebrates. The biological immune system is a parallel and distributed adaptive system that uses learning, memory, and associative recuperation to solve problems of recognition and classification. It particularly learns to recognize patterns, remember them, and use their combinations to build efficient pattern detectors. From the point of view of information processing, these interesting features are used

in the artificial immune system to successfully solve complex problems.

CONCLUSION

This article describes the main problems that are presently found in the field of bio-informatics. It also presents some of the bio-inspired computation techniques that provide solutions for problems related to classification, clustering, minimization, modelling, etc. The following article will describe a series of techniques that allow researchers to solve the above problems with bio-inspired models.

ACKNOWLEDGMENT

REFERENCES

Akatsu T, Miyano S and Kuhara S. (1999). *Identification of genetic networks from a small number of gene expression patterns under the boolean network model*. Proceedings of Pacific Symposium of Biocomputing 99:17-28.

Baldi P and Brunak S. (1998*). Bioinformatics: The machine Learning Approach*. MIT Press.

Beni G and Wang U. (1989). *Swarm Intelligence in cellular robotic systems*. NATO Advanced workshop on robots and biological systems. Il Ciocco Tuscany, Italy.

Bishop C.M. (1995*). Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

Bonabeau E, Dorigo M, Theraulaz G. (2001). *Swarm intelligence: From natural to artificial systems*. Journal of Artificial Societies and Social Simulation 4(1).

Dasgupta D. (1999). *Artificial immune system an their applications*. Springer-Verlang Berlin.

Domínguez E, Loza MI, Padín JF, Gesteira A, Paz E, Páramo M, Brenlla J, Pumar E, Iglesias F, Cibeira A, Castro M, Caruncho H, Carracedo A, Costas J. (2006). *Extensive linkage disequilibrium mapping at HTR2A and DRD3 for schizophrenia susceptibility genes in Galician population*. Schizophrenia Research, 2006.

Engelbrencht AP. (2005). *Fundamentals of computation swarm intelligence*. Wiley.

Farmer J, Pachard N and Parelson A. (1986). *The immune system, adaption and machine learning*. Physica D 2:189-204.

Fickett JW. (1996). *Finding genes by computer: The state of art*. Trends in Genetics 12(8):316:320.

Forbes N. (2004). *Imitation of Life. How Biology Is Inspiring Computing*. MIT Press.

Geard N. (2004). *Modelling Gene Regulatory Networks: Systems Biology to Complex Systems*. ACCS Draft Technical Report. ITEE Universisty of Queensland.

Holland J. (1975). *Adaption in Natual and Artificial Systems*. University of Michigan Press.

Hertz J., Krogh A. & Palmer RG. (1991). *Introduction to the theory of neural computation*. Addison-Wesley, Redwood City.

Koza J. (1990). *Genetic Programming: A paradigm for genetically breeding populations of computer programs to solve problems*. Stanford University Computer Science Department Technical Report.

Korf I, Yendel M and Bedell J.(2003). *Blast*. O'Relly.

Lopez-Bigas, N. & Ouzounis, C.A. (2004). *Genomewide identification of genes likely to be involved in human genetic disease*. Nucleic Acids Res. 32, 3108-14.

McCullock WS, Pitts W. (1943). *A Logical Calculus of Ideas Imminet in Nervous Activity*. Bulletin of Mathematical Biophysiscs 5:226-33.

B

Mullins, K.(1990). *The unusual origin of the polymerase chain reaction*. Scientific American 262(4):56-61.

Quackenbush J. (2001). *Computational Analysis of microarray data*. Nature Review Genetics 2:418-427.

Rechenberg I. (1973). *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD Thesis.

Rumelhart DE, Hinton GE & Williams RJ. (1986). *Learning Internal Representation by Backpropagation Errors*. Nature 323(99):533-6.

Setubal J and Meidanis J. (1999). *Introduction to Computational Molecular Biology*. PWS Publishing.

Smith TF and Waterman MS.(1981). *Identification of common molecular sequences*. Journal of Molecular Biology. 24(8):195-197.

Wallace IM, Blacsshields G and Higgins DG. (2005). *Multiple sequence alignments*. Current Opinion in Structural Biology. 15(3):231-267.

KEY TERMS

Amino Acid: One of the 20 chemical building blocks that are joined by amide (peptide) linkages to form a polypeptide chain of a protein.

Artificial Immune System: Biologically inspired computer algorithms that can be applied to various domains, including fault detection, function optimization, and intrusion detection. Also called *computer immune system*.

Electroforesis: The use of an external electric field to separate large biomolecules on the basis of their charge by running them through acrylamide or agarose gel.

Messenger RNA: The complementary copy of DNA formed from a single-stranded DNA template during the transcription that migrates from the nucleus to the cytoplasm where it is processed into a sequence carrying the information to code for a polypeptide domain.

Microarray: A 2D array, typically on a glass, filter, or silicon wafer, upon which genes or gene fragments are deposited or synthesized in a predetermined spatial order allowing them to be made available as probes in a high-throughput, parallel manner.

Nucleotid: A nucleic acid unit composed of a five carbon sugar joined to a phosphate group and a nitrogen base.

Swarm Intelligence: An artificial intelligence technique based on the study of collective behaviour in decentralised, self-organised systems.

Transcription: The assembly of complementary single-stranded RNA on a DNA template.

Translation: The process of converting RNA to protein by the assembly of a polypeptide chain from an mRNA molecule at the ribosome.

Bio–Inspired Algorithms in Bioinformatics II

José Antonio Seoane Fernández
University of A Coruña, Spain

Mónica Miguélez Rico
University of A Coruña, Spain

INTRODUCTION

Our previous article presented several computational models inspired on biological models, such as neural networks, evolutionary computation, swarm intelligence, and the artificial immune system. It also explained the most common problems in bioinformatics to which these models can be applied.

The present article presents a series of approaches to bioinformatics tasks that were developed by means of artificial intelligence techniques and focus on bio-inspired algorithms such as artificial neural networks and evolutionary computation.

BACKGROUND

Previous publications have focused on the use of bio-inspired and other artificial intelligence techniques. Keedwell (2005) has summarized the foundations of molecular biology, the main problems in bioinformatics, and the existing solutions based on artificial intelligence. Baldi (Baldi, 2001) also describes various techniques for problem-solving in bioinformatics. Other generalizing works on this subject can be found in (Larrañaga, 2006), whereas more specialized works focus on solutions based on evolutionary computation (Pal, 2006) or artificial life (Das, 2007).

Bio-Inspired Techniques

The following section describes how the techniques that were mentioned in our article *Bio-inspired Algorithms in Bioinformatics I* have been used to solve the main problems in bioinformatics.

Gene Expression

We start by describing how artificial intelligence techniques have contributed to the interpretation of genes expression. Artificial neural networks (ANNs) have been applied extensively to the classification of genetic data. One of the most commonly used architectures for the classification of this type of data is the multilayer perceptron. Many works use this architecture for diagnosis (Wang, 2006) (Wei, 2005) (Narayanan, 2004) and obtain very good results; most of these approaches use artificial neural networks to discover and classify interactions between variables (genes expression values).

Statnikov (2005) and Lee (2005) compare several classification techniques, such as ANNs using back-propagation, probabilistic ANNs, Support Vector Machines (SVM), K-Nearest Neighbour (KNN), and other statistical methods for the classification of data that issue from microarrays expression tests. In this type of genetic expression data classification, we can also find a combination of ANNs and genetic programming: Ritchie (Ritchie, 2004) codifies into each individual of the genetic algorithm (GA) the architecture and weights of the network, so that the genetic programming optimizes the network to minimize the error between the output layer and the expected output, or the hybrids between the ANNs and the genetic algorithms of Kim (Kim, 2004) and Keedwell (Keedwell, 2005).

Genetic programming (GP) as such has also been used (Gilbert, 2000; Hong, 2004; Langdon, 2004; Hong, 2006) to classify the results of an expression analysis. The advantage of GP is that it classifies the genes while selecting the relevant ones (Muni, 2006). The training set of the expression data *patients* and *control* are the input for the GP algorithm, which evaluates whether or not the example is a control. The result is one or a set of classification rules. The advantage of using GP instead of other techniques such as SVM is that it is transparent: the mechanism used to classify the examples of the patients can be evaluated (Driscoll, 2003).

Whereas the above studies all classify by means of supervised learning, the following section presents various expression analysis methods for clustering that

use non-supervised learning. This type of analysis is very useful to discover gene groups that are potentially related or associated to the illness. A comparison between the most commonly applied methods, using both real and simulated data, can be found in the works of Thalamuthu (2006), Handl (2005), and Sheng (2005). Even though these methods have provided good results in certain cases (Spellman, 1998; Tamayo, 1999; Mavroudi, 2002), some of their inherent problems, such as the identification of the number of clusters, the clustering of the "outliers", and the complexity associated to the large amount of data that are being analysed, often complicate their use for expression analysis (Sherlock, 2001). These deficiencies were tackled in a series of second generation clustering algorithms, among which the self-organising trees (Herrero, 2001; Hsu, 2003).

Another interesting approach for expression analysis is the use of the artificial immune system, which can be observed in the works of Ando (Ando 2003), who applies immune recognition to classification by making the system select the most significant genes and optimize their weights in order to obtain classification rules. Finally, de Sousa, de Castro, and Bezerra apply this technique to clustering (de Sousa, 2004)(de Castro, 2001)(Bezerra, 2003).

Sequence Alignment

Solutions based on genetic algorithms, such as the SAGA (Notredame, 1996), the RAGA, the PRAGA (Notredame, 1997, 2002), and others (O'Sullivan, 2004; Nguyen, 2002; Yokohama, 2001), have been applied to sequence alignment since the very beginning. The most common method consists in codifying the alignments as individuals inside the genetic algorithm. There are also hybrid solutions that use not only GA but also dynamic programming (Zhang, 1997, 1998); and finally, there is the application of artificial life algorithms, in particular the *ant colony* algorithm (Chen, 2006; Moss, 2003).

Genetic Networks

In order to correct the problem of the inference of genetic networks, the structure of the regulating network and the interactions between the participating genes must be predicted. The expression of the genes is regulated by transitions of states in which the levels of expression of the involved genes are updated simultaneously.

ANNs have been used to model these networks. Examples of such approaches can be found in the works of Krishna, Keedwell, and Narayanan (Keedwell, 2003)(Krishna, 2005).

Genetic algorithms (Ando, 2001)(Tominaga, 2001) and hybrid RNA-genetic approaches (Keedwell, 2005) have also been used for the same purpose.

Phylogenetic Trees

Normally, exhaustive search techniques for the creation of phylogenetic trees are computationally unfeasible for more than 10 comparisons, because the number of possible solutions increases exponentially with the number of objects in the comparisons. In order to optimize these searches, researchers have used heuristics based on genetic algorithms (Skourikhine, 2000)(Katoh, 2001)(Lemmon, 2002) that allow the reconstruction of the optimal trees with less computational load. Other techniques, such as the ant colony algorithm, have also been used to reconstruct phylogenetic trees (Ando, 2002)(Kummorkaew, 2004) (Perretto, 2005).

Gene Finding and Mapping

Gene mapping has been approached by methods that use only genetic algorithm (Fickett, 1996)(Murao, 2002) as well as by hybrid methods that combine genetic algorithms and statistical techniques (Gaspin, 1997).

The problem of gene searching and in particular promoter searching has been approached by means of neural networks (Liu, 2006), neural networks optimized with genetic algorithms (Knudsen, 1999), conventional genetic algorithms (Kel, 1998)(Levitsky, 2003),and fuzzy genetic algorithms (Jacob, 2005).

Structure Prediction

The tridimensional structure of DNA was predicted with genetic algorithms (Beckers, 1997) by codifying the torsional angles between the atoms of the DNA molecule as solutions of the genetic algorithm. Another approach was the development of hybrid strategies of ANNs and GAs (Parbhane, 2000), in which the network approaches the non-linear relations between the inputs and outputs of the data set, and the genetic algorithm searches within the network inputs space to optimize the output. In order to predict the secondary structure of the RNA, the system calculates the minimum free

energy of the structure for all the different combinations of the hydrogene links. There are approaches that use genetic algorithms (Shapiro, 2001)(Wiese, 2003) and artificial neural networks (Steeg, 1997).

Artificial neural networks have been applied to the prediction of protein structures (Qian, 1988)(Sasagawa, 1992), and so have genetic algorithms. A compilation of the application of evolutionary computation in protein structures prediction can be found in (Schulze-Kremer, 2000). Swarm intelligence, and optimization by ant colony in particular, have been applied to structures prediction (Shmygelska, 2005)(Chu, 2005) and artificial immune system (Nicosia, 2004)(Cutello, 2007).

CONCLUSION

This article presents a compendium of the most recent references on the application of bio-inspired solutions such as evolutionary computation, artificial neural networks, swarm intelligence, and artificial immune system to the most common problems in bioinformatics.

ACKNOWLEDGMENT

REFERENCES

Ando S and Iba H. (2001). *Inference of gene regulatory model by genetic algorithms*. Proceedings of Congress of Evolutionary computation. 1:712-719.

Ando S and Hiba H. (2002). *Ant algorithms for construction of evolutionary tree*. Proceedings of Congress of Evolutionary Computation (CEC 2002).

Ando S. and Iba H. (2003). *Artificial Immune System for Classification of Gene Expression Data*. Genetic

and Evolutionary Computation (GECCO 2003). LNCS 2724/2003 pp 205. Springer Berlin.

Baldi P and Brunak S. (2001). *Bioinformatics: The Machine Learning Approach*. MIT Press. 2001.

Beckers ML, Muydens LM, Pikkermaat JA and Altona C. (1997). *Aplications of a genetic algorithm in the conformational analysis of methylene acetal-linked thymine dimmers in DNA: Comparison with distance geometry calculations*. Jounal of Biomolecular NMR 9(1):25-34.

Bezerra GB and De Castro LN. (2003). *Bioinformatics data analysis using an artificial immune network*. In International Conference on Artificial Immune Systems 2003. LNCS 2787/2003 pp. 22-33. Springer Berlin.

Chen Y, Pan Y, Chen L and Chen J. (2006). *Partitioned Optimization Algorithms for multiple sequence alignment*. Second IEEE Workshop on High Performance Computing in Medicine and Biology (HiPCoMB-2006).

Chu D, Till M and Zomaya A. (2006). *Parallel ant colony optimization for 3D Protein Structure Prediction using HP Lattice Model*. 19th Congress on Evolutionary Computation (CEC 2006).

Cutello V, Nicosia G, Pavone M and Timmis J. (2007). *An immune algorithm for protein structure prediction on Lattice Models*. IEEE Transaction on Evolutionary Computation 11(1):101-117.

Das S, Abraham A and Konar A. (2007). *Swarm Intelligence Algorithms in Bioinformatics*. Computational Intelligence in Bioinformatics. Arpad, Keleman et al., editors. Springer Verlang Berlin.

De Smet F, Mathys J, Marchal K. (2002). *Adaptative quality based clustering of gene expression profiles*. Bioinformatics 20(5):660-667.

De Sousa JS, Gomes L, Bezerra GB, de Castro LN and Von Zuben FJ. (2004). *An immune evolutionary algorithm for multiple rearrangements of gene expression data*. Genetic Programming and Evolvable Machines. Vol 5 pp. 157-179.

De Castro LN & Von Zuben FJ. (2001). *aiNet: An artificial Immune Network for Data Analysis*. Data Mining: A Heuristic Approach. 2001 Idea Group Publishing.

Driscol JA, Worzel B and MacLean D. (2003). *Classification of gene expression data with genetic pro-*

gramming. Genetic Programming Theory and Practice. Kluwer Academic Publishers pp 25-42.

Fickett J and Cinkosky M. (1993). *A genetic algorithm for assembling chromosome physical maps.* Proceedings 2nd international conference in Bioinformatics, Supercomputing and Complex Genome Analysis 2:272-285. .

Gaspin C. and Schiex T. (1997). *Genetic Algorithms for genetic mapping.* Proceedings of 3rd European Conference in Artificial Evolution pp. 145-156.

Gilbert RJ, Rowland JJ and Kell DB. (2000). *Genomic computing: Explanatory modelling for functional genomics.* Proceedings of the Genetic and Evolutionary Computation conference (GECCO 2000). Morgan Kaufmann pp 551-557.

Handl J, Knowles J and Kell D. (2005). *Computational cluster validation in post-genomic data analysis.* Bioinformatics 21(15):3201-3212.

Herrero J, Valencia A and Dopazo J. (2001). *A hierarchical unsupervised growing neural network for clustering gene expression patterns.* Bioinformatics 17(2):126-162.

Hong JH and Cho SB. (2004). *Lymphoma cancer classification using genetic programming with SNR features.* Proceedings of EuroGP 2004. Coimbra pp78-88.

Hong JH and Cho SB. (2006). *The classification of cancer based on DNA microarray data that usere diverse ensemble genetic programming.* Artificial Intelligence in Medicine 36(1):43-58.

Hsu AL, Tang S and Halgamuge SK. (2003). *An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data.* Bioinformatics 19(16):2131-2140.

Jacob E, Sasikumar and Fair KNR. (2005). *A fuzzy guided genetic algorithm for operon prediction.* Bioinformatics 21(8):1403-1410.

Katoh K, Kuma K and Miyata T. (2005). *Genetic algorithm-based maximum likehood analysis for molecular phylogeny.* Journal of Molecular Biology. 53(4-5):477-484.

Keedwell E and Narayanan A. (2005). *Intelligent Bioinformatics.* Wiley.

Keedwell E and Narayanan A. (2005). *Discovering gene regulatory networks with a neural genetic hybrid.* IEE/ACM Transaction on Computational Biology and Bioinformatics. 2(3):231-243.

Kel A, Ptitsyn A, Babenko V, Meier-Ewert S and Lehrach H. (1998). *A genetic algorithm for designing gene family-specific oligonucleotide sets used for hybridization: The G protein-coupled receptor protein superfamily.* Bioinformatics 14(3):259-270.

Kim KJ and Cho SB. (2004). *Prediccion of colon cancer using an evolutionary neural network.* Neurocomputing 61:361-79.

Korf I, Yendel M and Bedell J. (2003). Blast. O'Relly.

Knudsen S. (1999). Promoter *2.0: for the recognition of Pol II promoter sequences.* Bioinformatics 15(5):356-417.

Krishna A, Narayanan A and Keedwell EC. (2005). *Neural netrowks and temporal gene expression data.* Applications of Evolutionary Computing (EVOBIO05) LNCS 3449 Springer Verlang.

Kummorkaew M, Ku K and Ruenglertpanyakul P. (2004). *Application of ant colony optimization to evolutionary tree construction.* Proceedings of 15th Annual Meeting of the Thai Society for Biotechnology. Thailand.

Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, Lozano J, Armañazas R, Santafé G, Pérez A and Robles V. (2006). *Machine Learning in Bioinformatics.* Briefings in Bioinformatics 7(1):86-112.

Langdon W and Buxton B. (2004). *Genetic programming for mining dna chip data from cancer patients.* Genetic Programming and Evolvable Machines. 5(3).

Lemmon AR and Milinkovitch MC. (2002). *The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation.* Proceedings of national academy of sciences. 99(16):10516-10521.

Lee JW, Lee JB, Park Mand Song SH. (2005). *An extensive comparison of recent classification tools applied to microarray data.* Journal of Computational Statistics and Data Analysis. 48(4):869-885.

Levitsky VG, Katokhin AV. (2003). *Recognition of eukaryotic promoters using genetic algorithm based*

on interactive discriminant analysis. In silico biology. 3(1-2):81-87.

Liu DR, Xiong X, DasGupta B, Zhang HG. (2006). *Motif discoveries in unaligned molecular sequences using self-organizing neural networks*, IEEE TRANS-ACTIONS ON NEURAL NETWORKS 17 (4): 919-928.

Mavroudi S, Papadimitriou S and Bezerianos A. (2002*). Gene expression data analysis with a dynamically extended self.-organized map that exploits class information*. Bioinformatics 18(11): 14446-1453.

Moss J and Johnson C. (2003). *An ant colony algorithm for multiple sequence alignment in bioinformatics*. Artificial Neural Networks and Genetic algorithms, pp 182-186. Springer.

Muni DP, Pal NR and Das J. (2006). *Genetic programming for simultaneous feature selection and classifier desing*. System, Man and Cybernetics 36(1):106-117.

Murao H, Tamaki H and Kitamura S. (2002). *A coevolutionary approach to adapt the genotype-phenotype map in genetic algorithms*. Proceedings of Congress of Evolutionary Computation 2:1612-1617.

Narayanan A, Keedwell E, Tatineni SS. (2004). *Single-layer artificial neural networks for gene expression analisys*. Neurocomputing 61:217-240.

Nguyen H, Yoshihara I, Yamamori K and Yusanaga M. *A parallel hybrid genetic algorithm for multiple protein sequence alignment*. Congress of Evolutionary Computation 1:309-314.

Nicosia G. (2004). *Immune Algorithms for Optimization and Protein Structure Prediction*. PhD Thesis. Department of Mathematics and Computer Science. University of Catania, Italy.

Notredame C and Higgins D. (1996). *SAGA: Sequence alignment by genetic algorithm*. Nucleic Acid Research. 24(8):1515-1524.

Notredame C, O'Brien EA and Higgins DG. (1997). *RAGA: RNA sequence alignment by genetic algorithm*. Nucleid Acid Research 25(22):4570-4580.

Notredame C. (2002). *Recent Progresses in multiple sequence alignment: a survey*. Pharmacogenomics 31(1); 131-144.

O'Sullivan O, Suhre K, Abergel C, Higgins D and Notredame C. (2004). *3DCoffee: Combining protein sequences and structures within multiple sequence alignments*. Journal of Molecular Biology 340(2):385-395.

Pal S, Bandyopadhyay S and Ray S. (2006). *Evolutionary Computation in Bioinformatics. A Review*. IEEE Transactions on System, Man and Cybernetics 36(5):601-615.

Parbhane R, Unniraman S, Tambe S, Nagaraja V and Kulkarni B. (2000). *Optimun DNA curvature DNA curvature using a hybrid approach involving an artificial neural network and genetic algorithm*. Journal of Biomolecular Structural Dynamics 17(4):665-672.

Perretto M and Lopes HS. (2005). *Reconstruction of phylogenetic trees using the ant colony optimization paradigm*. Genetic and Molecular research 4(3):581-589.

Prelic A, Bleuler S, Zimmermann P, Wille A, Bühlmann P, Gruissem W, Henning L, Thiele L and Zitzler E. (2006). *A systematic comparison and evaluation of bioclustering method for gene expression data*. Bioinformatics 22(9):1122-1129.

Qian N, Sejnowski TJ. (1988). *Predicting the secondary structure of globular proteins using neural network models*. Journal of Molecular Biology 202:865-884.

Ritchie MD, Coffey CS and Moore JH. (2004). *Genetics Programming Neural Networks as a Bioinformatics Tool for Human Genetics*. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004). LNCS 3012 Vol 2 pp438-448.

Sasegawa F. and Tajima K. (1992). *Prediction of protein secondary structures by a neural network*. Bioinformatics 9(2):147-152.

Schulze-Kremer S. (2000). *Genetic algorithms and protein folding. Methods in molecular biology*. Protein Structure Prediction: Methods and Protocols 143:175-222.

Shapiro BA, Wu JC, Bengali D and Potts MJ. (2001). *The massively parallel genetic algorithm for RNA folding: MIMD implementation and popular variation*. Bioinformatics 17(2):137-148.

Sheng Q, Moreau Y, De Smert G and Zhang MQ. (2005). *Advances in cluster analysis of microarray*

data. Data Analysis and Visualization in Genomics and Proteomics, John Wiley pp. 153-226.

Sherlock G. (2001). *Analysis of large-scale gene expression data.* Briefings in Bioinformatics 2(4):350-412.

Shmygelska A and Hoos H. (2005). *An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem*. BMC Bioinformatics 6:30.

Skourikhine A. (2000). *Phylogenetic tree reconstruction using self –adaptative genetic algorithm.* IEEE International Symposium in Bioinformatics and Biomedical engineering pp. 129-134.

Spellman PT, Sherlock G, Zhang MQ. (1998). *Comprehensive identification of cell cycleregulated genes of the yeast saccharomyces cerevisiase by microarray hybridization.* Molecular Biology Cell 9:3271-3378.

Statnikov A, Aliferis CF, Tsamardinos I. (2005). *A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis.* Bioinformatics 21(5):631-43.

Steeg E. (1997). *Neural networks, adaptative optimization and RNA secondary structure prediction.* Artificial Intelligence and Molecular Biology. MIT Press.

Tamayo P, Slonim D, Maserov J. (1999). *Interpreting patterns on gene expression with self-organizing maps: methods and application to hemotopoietics differectiation.* Proceedings of the National Academic of Sciences. 96:2907-2929.

Thalamuthu A, Mukhopadhyay I, Zheng X and Tseng G. (2006). *Evaluation and comparison of gene clustering methods in microarray analysis.* Bioinformatics 22(19):2405-2412.

Tominaga D, Okamoto M, Maki Y, Watanabe S and Eguchi Y. (1999). *Non-linear numeric optimization technique based on genetic algorithm for inverse problems: Towards the inference of genetic networks*. Computational Science and Biology (Proceedings of German Conference of Bioinformatics) pp 127-140.

Wang Z, Wang Y, Xuan J, Dong Y, Bakay M, Feng Y, Clarke R and Hoffman E. (2006). *Optimized multilayer perceptrons for molecular classification and diagnosis using genomic data*. Bioinformatics 22(6):755-761.

Wei JS. (2004). *Prediction of clinical outcome using gene expression profiling and artificial neural networks for patients with neuroblastoma.* Cancer Research 65:374.

Wiese KC and Glen E. (2003). A permutation-based genetic algorithm for the RNA folding problem: A Critical look of selection strategies, crossover operators, and representation issues. Biosystems 72(1-2):29-41.

Yokohama T, Watanabe T, Teneda A and Shimizu T. (2001). *A web server for multiple sequence alignment using genetic algorithm.* Genome Informatics. 12:382-283.

Zhang C and Wong AK. (1997). *Toward efficient multiple molecular sequence alignments: A system of genetic algorithm and dynamic programming.* IEEE transitions on System, Man and Cybernetics B. 27(6):918-932.

Zhang C and Wong AK. (1998). *A technique of genetic algorithm and sequence synthesis for multiple molecular sequence alignment.* Proc. Of IEEE transactions on System, Man and Cybernetics. 3:2442-2447.

KEY TERMS

Bioinformatics: The use of applied mathematics, informatics, statistics, and computer science to study biological systems.

Gene Expression: The conversion of information from gene to protein via transcription and translation.

Gene Mapping: Any method used for determining the location of a relative distance between genes on a chromosome.

Gene Regulatory Network: Genes that regulate or circumscribe the activity of other genes; specifically, genes with a code for proteins (repressors or activators) that regulate the genetic transcription of the structural genes and/or regulatory genes.

Phylogeny: The evolutionary relationships among organisms. The patterns of lineage branching produced by the true evolutionary history of the organism that is being considered.

Sequence Alignment: The result of comparing two or more gene or protein sequences in order to determine

their degree of base or amino acid similarity. Sequence alignments are used to determine the similarity, homology, function, or other degrees of relatedness between two or more genes or gene products.

Structure Prediction: Algorithms that predict the 2d or 3D structure of proteins or DNA molecules from their sequences.

B

Bioinspired Associative Memories

Roberto A. Vazquez
Center for Computing Research, IPN, Mexico

Humberto Sossa
Center for Computing Research, IPN, Mexico

INTRODUCTION

An **associative memory** AM is a special kind of neural network that allows recalling one output pattern given an input pattern as a key that might be altered by some kind of noise (additive, subtractive or mixed). Most of these models have several constraints that limit their applicability in complex problems such as **face recognition** (FR) and **3D object recognition** (3DOR).

Despite of the power of these approaches, they cannot reach their full power without applying new mechanisms based on current and future study of biological neural networks. In this direction, we would like to present a brief summary concerning a new associative model based on some neurobiological aspects of human brain. In addition, we would like to describe how this **dynamic associative memory** (DAM), combined with some aspects of **infant vision system**, could be applied to solve some of the most important problems of pattern recognition: FR and 3DOR.

BACKGROUND

Humans possess several capabilities such as learning, recognition and memorization. In the last 60 years, scientists of different communities have been trying to implement these capabilities into a computer. Along these years, several approaches have emerged, one common example are neural networks (McCulloch & Pitts, 1943) (Hebb, 1949) (Rosenblatt, 1958). Since the rebirth of neural networks, several models inspired in the neurobiological process have emerged. Among these models, perhaps the most popular is the feed-forward multilayer perceptron trained with the back-propagation algorithm (Rumelhart & McClelland, 1986). Other neural models are associative memories, for example (Anderson, 1972) (Hopfield, 1982) (Sussner, 2003) (Sossa, Barron & Vazquez, 2004). On the other hand,

the brain is not a huge fixed neural network as had been previously thought, but a dynamic, changing neural network. In this direction, several models have emerged for example (Grossberg, 1967) (Hopfield, 1982).

In most of these classical neural networks approaches, synapses are only adjusted during the training phase. After this phase, synapses are no longer adjusted. Modern brain theory uses continuous-time model based on current study of biological neural networks (Hecht-Nielse, 2003). In this direction, the next section described a new dynamic model based on some aspects of biological neural networks.

Dynamic Associative Memories (DAMs)

The dynamic associative model is not an iterative model as Hopfield's model. It emerges as an improvement of the model and results presented in (Sossa, Barron & Vazquez, 2007).

Let $\mathbf{x} \in \mathbf{R}^n$ and $\mathbf{y} \in \mathbf{R}^m$ an input and output pattern, respectively. An association between input pattern $\mathbf{x}$ and output pattern $\mathbf{y}$ is denoted as $(\mathbf{x}^k, \mathbf{y}^k)$, where k is the corresponding association. Associative memory: $\mathbf{W}$ is represented by a matrix whose components w_{ij} can be seen as the synapses of the neural network. If $\mathbf{x}^k = \mathbf{y}^k \forall k = 1, \ldots, p$ then $\mathbf{W}$ is auto-associative, otherwise it is hetero-associative. A distorted version of a pattern $\mathbf{x}$ to be recalled will be denoted as $\tilde{\mathbf{x}}$. If an associative memory $\mathbf{W}$ is fed with a distorted version of $\mathbf{x}^k$ and the output obtained is exactly $\mathbf{y}^k$, we say that recalling is robust.

Because of several regions of the brain interact together in the process of learning and recognition (Laughlin & Sejnowski, 2003), in the dynamic model there are defined several interacting areas; also it integrated the capability to adjust synapses in response to an input stimulus. Before the brain processes an input pattern, it is hypothesized that pattern is transformed and codified by the brain. This process is simulated

using the procedure introduced in (Sossa, Barron & Vazquez, 2004).

This procedure allows computing *coded patterns* and *de-coding patterns* from input and output patterns allocated in different interacting areas of the model. In addition a simplified version of $\mathbf{x}^k$ denoted by s_k is obtained as:

$$s_k = s\left(\mathbf{x}^k\right) = \mathbf{mid}\ \mathbf{x}^k \qquad (1)$$

where **mid** operator is defined as $\mathbf{mid}\ \mathbf{x} = x_{(n+1)/2}$.

When the brain is stimulated by an input pattern, some regions of the brain (interacting areas) are stimulated and synapses belonging to these regions are modified. In this model, the most excited interacting area is call *active region* (AR) and could be estimated as follows:

$$ar = r\left(\mathbf{x}\right) = \arg\left(\min_{i=1}^{p}\left|s\left(\mathbf{x}\right) - s_i\right|\right) \qquad (2)$$

Once computed the *coded patterns*, the *de-coding patterns* and s_k we can build the associative memory.

Let $\left\{\left(\overline{\mathbf{x}}^k, \overline{\mathbf{y}}^k\right) \middle| k = 1, \ldots, p\right\}$, $\overline{\mathbf{x}}^k \in \mathbf{R}^n$, $\overline{\mathbf{y}}^k \in \mathbf{R}^m$ a fundamental set of associations (coded patterns). Synapses of associative memory **W** are defined as:

$$w_{ij} = \overline{y}_i - \overline{x}_j \qquad (3)$$

In short, building of the associative memory can be performed in three stages as:

1. Transform the fundamental set of association into coded and de-coding patterns.
2. Compute simplified versions of input patterns by using equation 1.
3. Build **W** in terms of coded patterns by using equation 3.

There are synapses that can be drastically modified and they do not alter the behavior of the associative memory. On the contrary, there are synapses that can only be slightly modified to do not alter the behavior of the associative memory; we call this set of synapses *the kernel* of the associative memory and it is denoted by $\mathbf{K_W}$.

Let $\mathbf{K_W} \in \mathbf{R}^n$ the kernel of an associative memory **W**. A component of vector $\mathbf{K_W}$ is defined as:

$$kw_i = \mathbf{mid}\left(w_{ij}\right), j = 1, \ldots, m \qquad (4)$$

Synapses that belong to $\mathbf{K_W}$ are modified as a response to an input stimulus. Input patterns stimulate some ARs, interact with these regions and then, according to those interactions, the corresponding synapses are modified. An adjusting factor denoted by Δw can be computed as:

$$\Delta w = \Delta\left(\mathbf{x}\right) = s\left(\overline{\mathbf{x}}^{ar}\right) - s\left(\mathbf{x}\right) \qquad (5)$$

where *ar* is the index of the AR.

Finally, synapses belonging to $\mathbf{K_W}$ are modified as:

$$\mathbf{K_W} = \mathbf{K_W} \oplus \left(\Delta w - \Delta w_{old}\right) \qquad (6)$$

where operator $\oplus$ is defined as $\mathbf{x} \oplus e = x_i + e\ \forall i = 1, \ldots, m$.

Once synapses of the associative memory have been modified in response to an input pattern, every component of vector $\overline{\mathbf{y}}$ can be recalled by using its corresponding input vector $\overline{\mathbf{x}}$ as:

$$\overline{y}_i = \mathbf{mid}\left(w_{ij} + \overline{x}_j\right), j = 1, \ldots, n \qquad (7)$$

In short, pattern $\overline{\mathbf{y}}$ can be recalled by using its corresponding key vector $\overline{\mathbf{x}}$ or $\tilde{\mathbf{x}}$ in six stages:

1. Obtain index of the active region *ar* by using equation 2.
2. Transform $\mathbf{x}^k$ using de-coding pattern $\hat{\mathbf{x}}^{ar}$ by applying the following transformation: $\tilde{\mathbf{x}}^k = \mathbf{x}^k + \hat{\mathbf{x}}^{ar}$.
3. Compute adjust factor $\Delta w = \Delta\left(\hat{\mathbf{x}}\right)$ by using equation 5.
4. Modify synapses of associative memory **W** that belong to $\mathbf{K_W}$ by using equation 6.
5. Recall pattern $\hat{\mathbf{y}}^k$ by using equation 7.
6. Obtain $\mathbf{y}^k$ by transforming $\hat{\mathbf{y}}^k$ using de-coding pattern $\hat{\mathbf{y}}^{ar}$ by applying transformation: $\mathbf{y}^k = \hat{\mathbf{y}}^k - \hat{\mathbf{y}}^{ar}$.

The formal set of propositions that support the correct functioning of this dynamic model, the main advantages against other classical models and some interesting applications of this model are described in (Vazquez, Sossa & Garro, 2006) and (Vazquez & Sossa, 2007).

In general, we distinguish two main parts in this model: a part concerning to the determination of the AR (PAR) and a part concerning to pattern recall (PPR). PAR (first step during recall procedure) sends a signal to PPR (remaining steps for recall procedure) and indicates the region activated by the input pattern.

FACE AND 3D OBJECT RECOGNITION USING SOME ASPECTS OF THE INFANT VISION SYSTEM AND DAMS

Several statistical computationally expensive techniques (dimension reduction techniques) such as principal component analysis and factor analysis have been proposed, for solving the FR and 3DOR problem.

Instead of using the complete version of the describing pattern *X* of any face or object, a simplified version from describing pattern *X* could be used to recognize a face or an object. In many papers, authors have used PCA to perform FR and other tasks, refer for example to (Turk & Pentland, 1991).

During early developmental stages, there are communication pathways between the visual and other sensory areas of the cortex, showing how the biological network is self-organizing. Within a few months of birth, the baby is able to differentiate one face or objects (toys) from others. Barlow hypothesized that for a neural system one possible way of capturing the statistical structure was to remove the redundancy in the sensory outputs (Barlow, 2001).

By taking into account the theory of Barlow, we propose a novel method for FR and 3DOR based on some biological aspects of infant vision. The biological hypotheses of this proposal are based on the role of the response to **low frequencies** at early stages, and some conjectures concerning how an infant detects subtle features (**stimulating points** (SP)) in a face or object (Mondloch et al., 1999; Acerra, Burnod, & Schonen, 2002).

The proposal consists on several DAMs used to recognize different images of faces and objects. As the infant vision responds to low frequencies of the signal, a low-pass filter is first used to remove high frequency components from the image. After that, we divide the image in different parts (sub-patterns). Then, over each sub-pattern, we detect subtle features by means of a random selection of SPs. Preprocessing images used to remove high frequencies and random selection of SPs contribute to eliminating redundant information and help the DAMs to learn efficiently the faces or the objects. At last, each DAM is fed with these sub-patterns for training and recognition.

Response to Low Frequencies

Instead of using a filter that exactly simulates the infant vision system behavior at any stage, we use a low-pass filter to remove high frequency. This kind of filter could be seen as a slight approximation of the infant vision system due to it eliminates high frequency components from the pattern, see Figure 1.

Random Selection

In the DAM model, the simplified version of an input pattern is the middle value of input pattern. In order to simulate the random selection of the infant vision system we have substituted **mid** operator with **rand** operator defined as follows:

$$\mathbf{rand\ x} = x_{sp} \tag{8}$$

where $sp = random(n)$ is a random number between zero and the length of input pattern. sp is a constant value computed at the beginning of the building phase and represents a SP. During recalling phase sp takes the same value.

rand operator uses a uniform random generator to select a component over each part of the pattern. We adopt this operator based on the hypothetical idea about infants are interested into sets of features where each set is different with some intersection among them. By selecting features at random, we conjecture that at least we select a feature belonging to these sets.

Implementation of the Proposal

During recalling, each DAM recovers a part of the image based on the AR of each DAM. However, a part of the image could be wrongly recalled because its

Figure 1. Images filtered with masks of different size. Each group could be associated with different stages of infant vision system.

B

corresponding AR could be wrongly determined due to some patterns do not satisfy the prepositions that guarantee perfect recall. To avoid this, we use an integrator. Each DAM determines an AR, the index of the AR is sent to the integrator, the integrator determines which was the most voted region and sends to the DAMs the index of the most voted region (the new AR).

Let $\left[\mathbf{I}_x^k\right]_{a\times b}$ and $\left[\mathbf{I}_y^k\right]_{c\times d}$ an association of images and *r* be the number of DAMs. Building of the nDAMs is done as follows:

1. Select filter size and apply it to the images.
2. Transform the images into a vector ($\mathbf{x}^k, \mathbf{y}^k$) by means of the standard image scan method where vectors are of size $a\times b$ and $c\times d$ respectively.
3. Decompose $\mathbf{x}^k$ and $\mathbf{y}^k$ in *r* sub-patterns of the same size.
4. Take each sub-pattern (from the first one to the last one (*r*)), then take at random a SP $sp_i, i=1,\ldots,r$ and extract the value at that position.
5. Train *r* DAMS as in building procedure taking each sub-pattern (from the first one to the last one (*r*)) using **rand** operator.

Pattern $\mathbf{I}_y^k$ can be recalled by using its corresponding key image $\mathbf{I}_x^k$ or $\tilde{\mathbf{I}}_x^k$ as follows:

1. Select filter size and apply to the images.
2. Transform the images into a vector by means of the standard image scan method and decompose $\mathbf{x}^k$ in *r* sub-patterns of the same size.
3. Use the SP, $sp_i, i=1,\ldots,r$ computed during the building phase and extract the value of each sub-pattern.
4. Determine the most voted active region using the integrator.
5. Substitute **mid** with **rand** operator in recalling procedure and apply steps from two to six as described in recalling procedure on each DAM.
6. Finally, put together recalled sub-patterns to form the output pattern.

A schematic representation of the building and recalling phases is shown in Figure 2.

Some Experimental Results

To test the accuracy of the proposal, we performed two experiments. In experiment 1, we used a benchmark (Spacek, 1996) of faces of 15 different people. In experiment 2, we use a benchmark (Nene, 1996) of 100 objects. During the training process in both experiments, the DAM performed with 100% accuracy using only one image of each person and object. During testing, the DAM performed in average with 99% accuracy for the remaining 285 images of faces (experiment 1) and 95% accuracy for the remaining 1900 images of

Figure 2. (a) Schematic representation of building phase. (b) Schematic representation of the recalling phase.

(a)

(b)
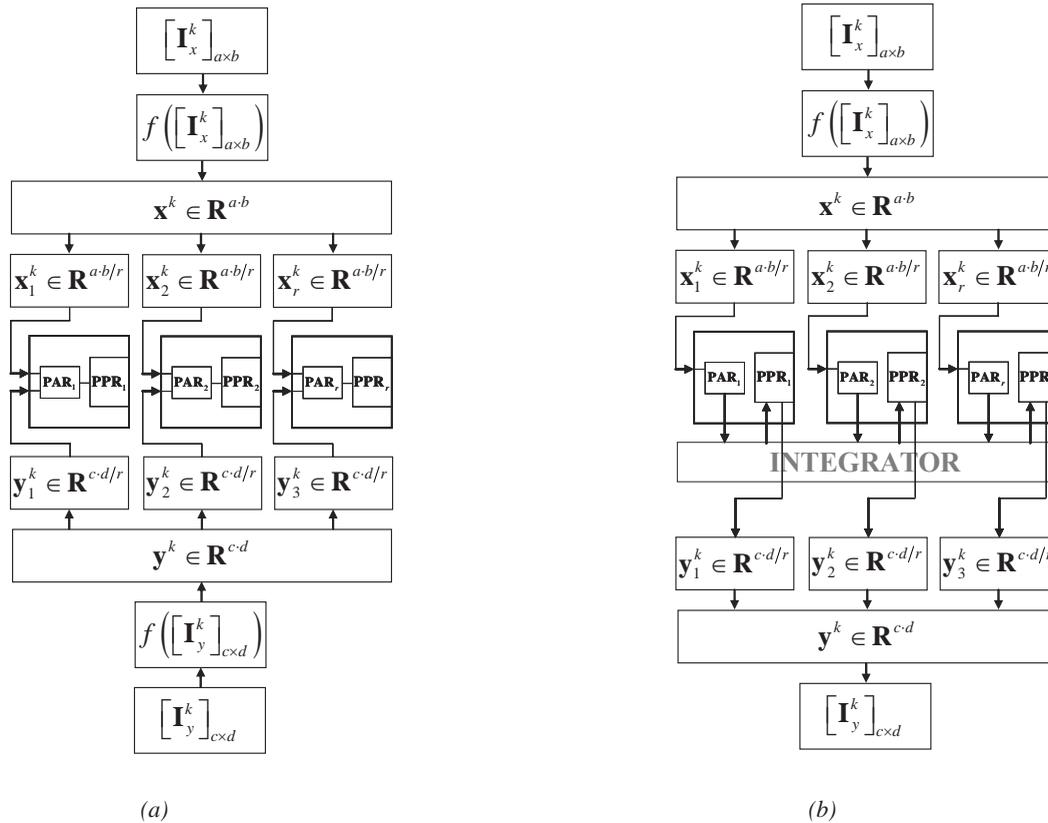
objects (experiment 2) by using different sized-filter and SPs.

Through several experiments we have tested the accuracy and stability of the proposal using different number of stimulation points, see Figure 3 and Figure 4. Because of SPs (pixels) were randomly selected, we decided to test the stability of proposal with the same configuration 20 times.

An extra experiment was performed with images partially occluded. In average, the accuracy of the proposal diminished to 80%.

While PCA dimension reduction techniques require the covariance matrix to build an Eigenspace, then to project patterns using this space to eliminate redundant information, our proposal only requires removing high frequencies by using a filter and a random selection of stimulating points.

This approach contributes to eliminating redundant information; it is less computationally expensive than PCA, and helps the DAMs or other classification tools to learn efficiently the faces or objects.

FUTURE TRENDS

Preprocessing images used to remove high frequencies and random selection of SPs contribute eliminating unnecessary information and help the DAM to learn efficiently faces and objects. Now we need to study new mechanisms based on evolutionary techniques in order to select the most important SPs. In addition, we need to test different types of filters that really simulate the behavior of the infant vision system.

In a near future, we pretend to use this proposal as a biological model to explain the learning process in infant's brain for FR and 3DOR. One step in this direction can be found in (Vazquez & Sossa, 2007).

CONCLUSION

In this paper, we have proposed a novel method for FR and 3DOR based on some biological aspects of infant vision. We have shown that by applying some aspects of the infant vision system it is possible to enhance the performance of an associative memory (or other

Figure 3. Accuracy of the proposal using different filter size. The reader can verify the accuracy of the proposal diminish after apply a filter of size greater than 25.
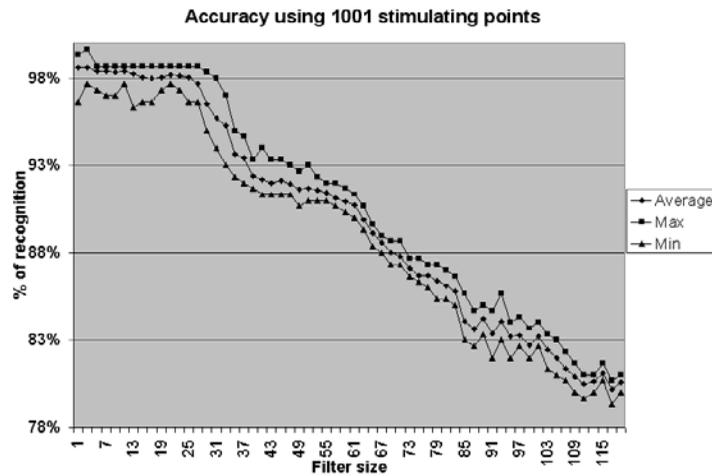
Figure 4. Average accuracy of the proposal. Maximum, average and minimum accuracy are sketched.
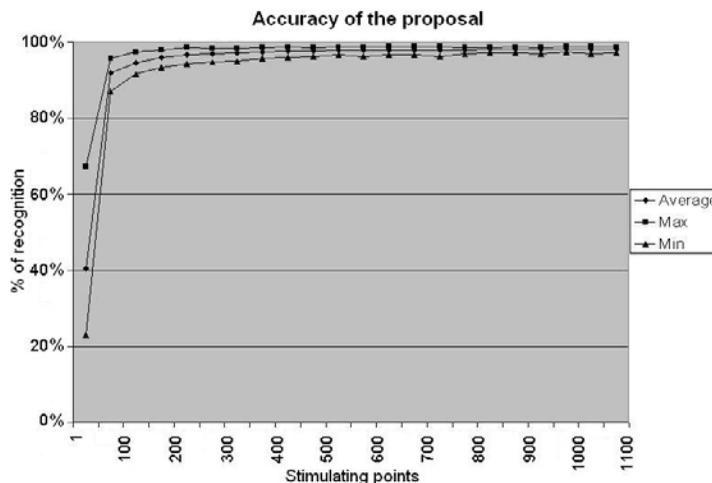
distance classifiers) and make possible its application to complex problems such as FR and 3DOR.

In order to recognize different images of face or objects we have used several DAMs. As the infant vision responds to low frequencies of the signal, a low-filter is first used to remove high frequency components from the image. Then we detected subtle features in the image by means of a random selection of SPs. At last, each DAM was fed with this information for training and recognition.

Through several experiments, we have shown the accuracy and the stability of the proposal even under occlusions. In average, the accuracy of the proposal oscillates between 95% and 99%.

The results obtained with the proposal were comparable with those obtained by means of a PCA-based method (99%). Although PCA is a powerful technique it consumes a lot of time to reduce the dimensionality of the data. Our proposal, because of its simplicity in operations, is not a computationally

expensive technique and the results obtained are comparable to those provided by PCA.

REFERENCES

Acerra, F., Burnod, Y. & Schonen, S. (2002). Modelling aspects of face processing in early infancy. *Developmental science*, 5(1), 98-117.

Anderson, J. A. (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences*, 14(3-4), 197-220.

Barlow, H. B. (2001). Redundancy Reduction Revisited. *Network: Computation in Neural Systems*, 12:241-253.

Grossberg, S. (1967). Nonlinear difference-differential equations in prediction and learning theory. *Proceedings of the National Academy of Sciences*, 58(4), 1329–1334.

Hebb, D. O. (1949). *The Organization of Behavior*, New York: Wiley.

Hecht-Nielse, *et al.* (2003). A theory of the thalamo-cortex. *Computational models for neuroscience*, pp 85-124, Springer-Verlag, London.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554-2558.

Laughlin, S. B. & Sejnowski, T. J. (2003). Communication in neuronal networks. *Science,* 301(5641), 1870-1874.

McCulloch, W.S. & Pitts, W.H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical biophysics*, 5(1-2), 115–133.

Mondloch, C. J. *et al.* (1999). Face perception during early infancy. *Psychological Science*, 10(5), 419-422.

Nene, S. A. *et. al.* (1996). Columbia Object Image Library (COIL 100). *Technical Report* No. CUCS-006-96. Department of Computer Science, Columbia University.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.

Rumelhart, D. & McClelland, J. (1986). Parallel distributed processing group. MIT Press.

Sossa, H., Barron, R. & Vazquez, R. A. (2004). Transforming fundamental set of patterns to a canonical form to improve pattern recall. *Lecture Notes in Artificial Intelligence* 3315, 687-696.

Sossa, H., Barron, R. & Vazquez, R. A. (2007). Study of the influence of noise in the values of a median associative memory. *Lecture Notes in Computer Sciences*, 4432, 55-62.

Spacek, L. (1996). Collection of facial images: Grimace. Available from http://cswwww.essex.ac.uk/mv/allfaces/grimace.html

Sussner, P. (2003). Generalizing operations of binary auto-associative morphological memories using fuzzy set theory. *Journal of Mathematical Imaging and Vision,* 19(2), 81-93.

Turk, M. & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 71-86.

Vazquez, R. A., Sossa, H. & Garro, B. A. (2006). A new bi-directional associative memory. *Lecture Notes in Artificial Intelligence*, 4293, 367-380.

Vazquez, R. A. & Sossa H. (2007). A computational approach for modeling the infant vision system in object and face recognition. *Journal BMC Neuroscience* 8(suppl 2), P204.

KEY TERMS

Associative Memory: Mathematical device specially designed to recall output patterns from input patterns that might be altered by noise.

Dynamic Associative Memory: A special type of associative memory composed by dynamical synapses. This memory adjusts the values of their synapses during recalling phase in response to input stimuli.

Dynamical Synapses: Synapses that modified their values in response to an input stimulus also during recalling phases.

Low-Pass Filter: Filter which removes high frequencies from an image or signal. This type of filters is used to simulate the infant vision system at early stages. Examples of these filters are the average filter or the median filter.

PCA: Principal component analysis is a technique used to reduce multidimensional data sets to lower dimensions for analysis. PCA involves the computation of the eigenvalue decomposition of a data set, usually after mean centering the data for each attribute.

Random Selection: Selection of one or more components of a vector at randomly manner. Random selection techniques are used to reduce multidimensional data sets to lower dimensions for analysis.

Stimulating Points: Characteristic points of an object in an image used during learning and recognition, which capture the attention of a child. These stimulating points are used to train the dynamic associative memory.

Bio-Inspired Dynamical Tools for Analyzing Cognition

Manuel G. Bedia
University of Zaragoza, Spain

Juan M. Corchado
University of Salamanca, Spain

Luis F. Castillo
National University, Colombia

INTRODUCTION

The knowledge about higher brain centres in insects and how they affect the insect's behaviour has increased significantly in recent years by theoretical and experimental investigations. Nowadays, a large body of evidence suggests that higher brain centres of insects are important for learning, short-term, long-term memory and play an important role for context generalisation (Bazhenof *et al*., 2001). Related to these subjects, one of the most interesting goals to achieve would be to understand the relationship between sequential memory encoding processes and the higher brain centres in insects in order to develop a general "insect-brain" control architecture to be implemented on simple robots. In this contribution, it is showed a review of the most important and recent results related to spatio-temporal coding and it is suggested the possibility to use continuous recurrent neural networks (CRNNs) (that can be used to model non-linear systems, in particular Lotka-Volterra systems) in order to find out a way to model simple cognitive systems from an abstract viewpoint. After showing the typical and interesting behaviors that emerge in appropriate Lotka-Volterra systems (in particular, *winnerless competition processes*) next sections deal with a brief discussion about the intelligent systems inspired in studies coming from the biology.

BACKGROUND

What do we name "computation"? Let us say a system shows the capability to compute if it has memory (or some form of internal plasticity) and it is able to determine the appropriate decision (or behavior, or action) given a criteria and making calculations using what it senses from the outside world. Some biological systems, like several insects, have brains that show a type of computation that may be described functionally by a specific type of non-linear dynamical systems called Lotka-Volterra systems (Rabinovich *et al*., 2000). According to our objectives, one of the first interests focuses on how an artificial recurrent neural network could model a non-linear system, in particular, a Lotka-Volterra system (Afraimovich *et al*., 2004) and what are the typical processes that emerge in Lotka-Volterra systems (Rabinovich *et al*., 2000). If it could be understood, then it would be clearer how the relationships between sequential memory encoding processes and the higher brain centres in insects are.

About higher brain centers (and how they affect an insect's behaviour) it is possible to stop the functioning of particular neurons under investigation during phases of experiments and gradually reestablish the functioning of the neural circuit (Gerber *et al*., 2004). At the present, it is known that higher brain centers in insects are related on autonomous navigation, multi-modal sensory integration, and to an insect's behavioral complexity generally; evidence also suggests an important role for context generalization, short-term and long-term memory (McGuire *et al*., 2001). For a long time, insects have inspired robotic research in a qualitative way but insect nervous systems have been under-exploited as a source for potential robot control architectures. In particular it often seems to be assumed that insects only perform 'reactive' behavior, and more complex control will need to be modeled on 'higher' animals.

B

SPATIO-TEMPORAL NEURAL CODING GENERATOR

The ability to process sequential information has long been seen as one of the most important functions of "intelligent" systems (Huerta *et al*., 2004). As it will be shown afterwards, winnerless competition principle appears as a major type of mechanism of sequential memory processing. The underlying concept is that sequential memory can be encoded in a (multidimensional) dynamical system by means of heteroclinic trajectories connecting several saddle points. Each of the saddle points is assumed to be remembered for further action (Afraimovich et al., 2004).

Computation over Neural Networks

Digital computers are considered universal in the sense of capability to implement any symbolic algorithm. If artificial neural networks, that have a great influence on the field of computation, are considered as a paradigm of computation, one may ask how the relation between neural networks and the classical computing paradigm is. For this question it is needed to consider, on the one hand, discrete computation (digital) and on the other hand, nondiscrete computation (analog). In terms of the first, the traditional paradigm is the Turing Machine with the Von Neumann architecture. A decade ago it was shown that artificial neural networks of analog neurons and rational weights are computationally equivalent to Turing machines. In terms of analog computation, it was also showed that three-layer feedforward nets can approximate any smooth function with arbitrary precision (Hornik *et al*., 1990). This result was extended to show how continuous recurrent neural nets (CRNN) can approximate an arbitrary dynamical system as given by a system of *n* coupled first-order differential equations (Tsung, 1994; Chow and Li, 2000).

Neural Network Computation from a Dynamical-System Viewpoint

Modern dynamical systems theory is concerned with the qualitative understanding of asymptotic behaviors of systems that evolve in time. With complex non-linear systems, defined by coupled differential, difference or functional equations, it is often impossible to obtain closed-form (or asymptotically closed form) solutions. Even if such solutions are obtained,

their functional forms are usually too complicated to give an understanding of the overall behavior of the system. In such situations qualitative analysis of the limit sets (fixed points, cycles or chaos) of the system can often offer better insights. Qualitative means that this type of analysis is not concerned with the quantitative changes but rather what the limiting behavior will be (Tsung, 1994).

Spatio-Temporal Neural Coding and Winnerless Competition Networks

It is important to understand how the information is processed by computation from a dynamical viewpoint (in terms of steady states, limit cycles and strange attractors) because it gives us the possibility of manage sequential processes (Freeman, 1990). In this section it is showed a new direction in information dynamics namely the Winnerless Competition (WLC) behavior. The main point of this principle is the transformation of the incoming spatial inputs into identity-temporal output based on the intrinsic switching dynamics of a dynamical system. In the presence of stimuli the sequence of the switching, whose geometrical image in the phase space is a heteroclinic contour, uniquely depends on the incoming information.

Consider the generalized Lotka-Volterra system (N=3):

$$\dot{a}_1 = a_1\left[1 - \left(a_1 + \rho_{12}a_2 + \rho_{13}a_3\right)\right]$$
$$\dot{a}_2 = a_2\left[1 - \left(a_2 + \rho_{21}a_1 + \rho_{23}a_3\right)\right]$$
$$\dot{a}_3 = a_3\left[1 - \left(a_3 + \rho_{31}a_2 + \rho_{32}a_2\right)\right]$$

If the following matrix and parameter conditions are satisfied,

$$\left(\rho_{ij}\right) = \begin{pmatrix} 1 & \alpha_1 & \beta_1 \\ \beta_2 & 1 & \alpha_2 \\ \alpha_3 & \beta_3 & 1 \end{pmatrix}$$

$$0 < \alpha_i < 1 < \beta_i$$

When the coefficients fulfill that $\alpha_1 = \alpha_2 = \alpha_3 < 1$ and $\beta_1 = \beta_2 = \beta_3 > 1$, we have three cases:

1. Stable equilibrium with all three components simultaneously present/working.
2. Three equilibria (1,0,0), (0,1,0) and (0,0,1) all stable, each one attainable depending on initial conditions.

Figure. 1. Topology of the behavior in the phase space of WLC neurons net in a 3D. The axes represent the addresses of the oscillatory connections between the processing units (α = 0.5, β = 1.8, x(0) = 1.01, y(0) = 1.01, z(0) = 0.01).
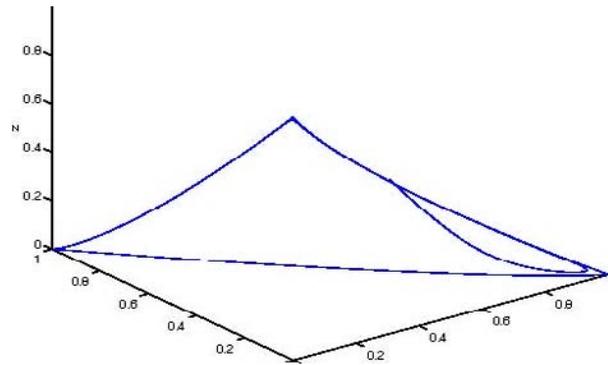
3. Neither equilibrium points nor periodic solutions are asymptotically stable and we have wandering trajectories defining Winnerless Competition (WLC) behaviour

The advantages of dealing with Lotka-Volterra systems are important. It has been shown above how a Winnerless competition process can emerge in a generalized Lotka-Volterra system. Also it is known that this type of process is generalizable to any dynamical system and that any dynamical system can be represented by using recurrent neural networks (Hornik *et al*., 1990). From this point of view, winnerless competition processes can be obtained whenever that exits a boundary condition: the Lotka-Volterra system must be of any dimension n greater than three to find Winnerless competition behavior. In the following, it is assumed that Lotka-Volterra systems approximate arbitrarily closely the dynamics of any finite-dimensional dynamical system for any finite time and we will assume and concentrate in showing them as a type of neural nets with great interest for applications (Hopfield, 2001). Various attempts at modeling the complex dynamics in insect brains have been made (Nowotny *et al*., 2004; Rabinovich *et al*. 2001), and it is suggested that simple CRNN systems (Continuous and recurrent neural network) could be an alternative framework to implement competing processes between neurons that generate spatio-temporal patterns to codify memory in a similar way simplest living systems do (Rabinovich *et al*. 2006). Recurrent neural networks of competing neuron (inspired in how higher brain centres in insects

work) would allow to explore how building sequential memory and might suggest control architectures of insect-inspired robotic systems.

Winnerless Competition Systems Generate Adaptive Behavior

Some features of the winnerless competition systems seem to be very promising to use these systems to model the activity and the design of intelligent artefacts. It is focused on some of the results of previous theoretical studies of some authors on systems of n elements coordinated with excitement-inhibition relations (Rabinovich *et al*., 2001). These systems show:

•   **Large Capacity:** A heteroclinic (spatiotemporal) representation provides greatly increased capacity to the system. Because sequences of activity are combinatorial across elements and time, overlap between representations can be reduced, and the distance in phase space between orbits can be increased.

•   **Sensitivity (to similar stimulus) and, simultaneously, capacity for categorization:** This is because the heteroclinic linking of a specific set of saddle points is always unique. Two like stimuli, activating greatly overlapping subsets of a network, may become easily separated because small initial differences will become amplified in time.

•   **Robustness:** In the following sense, the attractor of a perturbed system remains in a small neighbor-

hood of the "unperturbed" attractor (robustness as topological similarity of the perturbed pattern).

All these important features emerge from the dynamic of the Lotka-Volterra system. There are more examples: in [Nepomnyashchikh *et al*, 2003] is described a simple chaotic system of coupled oscillators that shows a complex and fruitful adaptive behaviour; the interaction among the activity of elements in the model and external inputs give rise to an emergence of searching rules from basic properties of nonlinear systems (rules which have not been pre-programmed explicitly) and with obvious adaptive value. More in detail: the adaptive rules are autonomous (the system selects an appropriate rule with no instructions from outside), and they are the result of interaction between intrinsic dynamics of the system and dynamics of the environment. These rules emerge, in a spontaneous way, because of the non-linearity in the simple system.

Winnerless Competition for Computing and Interests in Robotics

The suggestion of using heteroclinic trajectories with computing purposes shows advantages for robotics interests. It is known that very simple dynamical systems are equivalent to Turing machines and also that computing with heteroclinic orbits adds to the classical computing the feature of high sensitivity to initial conditions increasing. If we consider artefacts with computation processes ordered by winnerless competition behaviour, the artefacts will have great ability to process, manage and store sequential information. In spite of the history of studies of sequential learning and memory, little is known about dynamical principles of storing and remembering of multiple events and their temporal order by neural networks. This principle called winnerless competition can be a very useful mechanism to explore and model sequential and scheduled processes in industrial and robotic problems.

FUTURE TRENDS

Computation by heteroclinic orbits provides new perspectives to traditional computing. Because of its features, it could be interesting building such a kind of bio-inspired systems based in Winnerless competition processes. Evolution has chosen the nonlinear dynamical phenomena as the basis of the adaptive behaviour patterns of the living organisms and these systems show, in one hand, the coexistence of sensitivity (ability to distinguish distinct, albeit similar, inputs) and robustness (ability to classify similar signals receptions as the same one). If we are able to reproduce the same characteristics in artificial intelligent architectures, will make it easier to go beyond the actual limitations into the intelligent systems applied to the real problems.

CONCLUSION

It has been summarized how a system architecture whose stimulus-dependent dynamics reproduces spatio-temporal features could be able to code and build a memory inspired in the higher brain centres of insects (Nowotny *et al*., 2004). Beyond the biological observations which suggested these investigations, recurrent neural networks where winnerless competition processes can emerge, provide an attractive model for computation because of their large capacity as well as their robustness to noise contamination. It has been showed an interesting tool (using control and synchronization of spatio-temporal patterns) to transfer and process information between different neural assemblies for classification problems in, eventually, several industrial environments. For example, winnerless competition processes could be able to solve the fundamental contradiction between sensitivity and generalizing of the recognition, multistability and robustness to the noise in real processes (Rabinovich *et al*., 2000). For classification tasks, is useful to get models that could be reproducible. In the language of non-linearity, this is possible only if the system is strongly dissipative (in other words, if it can rapidly forget its initial state). On the other hand, a useful classificator system should be sensitive to small variations in the inputs, so that fine discriminations between similar but not identical stimuli are possible. Winnerless competition principle shows both features.

REFERENCES

Afraimovich, V.S., Rabinovich, M.I. and Varona, P. (2004). Heteroclinic contours in neural ensembles and the winnerless competition principle. International Journal of Bifurcation and Chaos, 14: p. 1195-1208.

Bazhenof, M., Stopfer, M., Rabinovich, M., Abarbanel, H., Sejnowski, T. J. and Laurent, G. (2001) Model of cellular and network mechanisms for odor-evoked temporal patterning in the locust antennal lobe. Neuron 30, 569-581 (2001)

Chow T.M., Ho, J.K.L and Li, X.D. (2005). Approximation of dynamical time-variant systems by continuous-time recurrent neural networks. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 52, No. 10, pp. 656 - 660.

Freeman, W.J. and Holmes, M.D. (2005). Metastability,instability,and state transition in neocortex. Neural Networks 18(5-6): pp. 497-504

Gerber, B., Tanimoto, H., and Heisenberg, M. (2004). An engram found? Evaluating the evidence from fruities. Current Opinion in Neurobiology, 14:737-768.

Hopfield, J. J. and Brody, C. D. (2001). What is a moment? Transient synchrony as a collective mechanism for spatio-temporal integration. Proceedings of the National Academy of Science, vol. 98, Issue 3, p.1282-1287

Hornik, K., Stinchcombe, M. and White, H. (1990) Universal approximation of an unknown mapping and its derivates using multilayer feedforward networks. Neural Networks, 3:551-560 (1990).

Huerta, R. and Rabinovich, M. (2004). Reproducible sequence generation in random neural ensembles. Physical Review Letters, 93 (23), 238104-1-4

Tsung, Fu-Sheng (1994). Modeling Dynamical Systems with Recurrent Neural Networks. PhD thesis, Department of Computer Science. University of California, San Diego, 1994

Nowotny, T., Huerta, R., Abarbanel, H. D. I. and Rabinovich, M. I. (2004). Learning classification and association inthe olfactory system of insects. Neural Computation Vol. 16, Issue 8 - pp. 1601 - 1640

McGuire, S., Le, P., and Davis, R. (2001). The role of Drosophila mushroom body signaling in olfactory memory. Science, 293:1330-1333.

Nepomnyashchikh, V., Podgornyj K. (2003) Emergence of Adaptive Searching Rules from the Dynamics of a Simple Nonlinear System. Adaptive Behavior, 11(4): 245-265.

Rabinovich, M.I., Varona, P. and Abarbanel, H. (2000). Nonlinear cooperative dynamics of living neurons. Int. J. .Bifurcation Chaos 10 (5), 913-933

Rabinovich, M.I., Volkovskii, A., Lecanda, P., Huerta, R., Abarbanel, H., and Laurent, G. (2001). Dynamical encoding by networks of competing neuron groups: winnerless competition. Physical Review Letters, 87:068102(4).

Rabinovich M.I., Huerta R., Afraimovich V.(2006). Dynamics of sequential decision making. Physical Review Letters, 97(18): 188103.

KEY TERMS

Adaptive Behaviour: Type of behavior that allows an individual to substitute a disruptive behavior to something more constructive and able to adapt to a given situation.

Bio-Inspired Techniques: Bio-inspired systems and tools are able to bring together results from different areas of knowledge, including biology, engineering and other physical sciences, interested in studying and using models and techniques inspired from or applied to biological systems.

Computational System: Computation is a general term for any type of information processing that can be represented mathematically. This includes phenomena ranging from simple calculations to human thinking. A device able to make computations is called computational system.

Dynamical Recurrent Networks: Complex nonlinear dynamic system described by a set of nonlinear differential or difference equations with extensive connection weights.

Heteroclinic Orbits: In the phase portrait of a dynamical system, a heteroclinic orbit (sometimes called a heteroclinic connection) is a path in phase space which joins two different equilibrium points. If the equilibrium points at the start and end of the orbit are the same, the orbit is a homoclinic orbit.

Stability-Plasticity Dilemma: It explores how a learning system remains adaptive (plastic) in response to significant input, yet remains stable in response to irrelevant input.

Winnerless Competition Process: Dynamical process whose main point is the transformation of the incoming identity or spatial inputs into identity-temporal output based on the intrinsic switching dynamics of the neural system.

Biometric Security Technology

Marcos Faundez-Zanuy
Escola Universitària Politècnica de Mataró, Spain

INTRODUCTION

The word biometrics comes from the Greek words "bios" (life) and "metrikos" (measure). Strictly speaking, it refers to a science involving the statistical analysis of biological characteristics. Thus, we should refer to biometric recognition of people, as those security applications that analyze human characteristics for identity verification or identification. However, we will use the short term "biometrics" to refer to "biometric recognition of people".

Biometric recognition offers a promising approach for security applications, with some advantages over the classical methods, which depend on something you have (key, card, etc.), or something you know (password, **PIN**, etc.). A nice property of biometric traits is that they are based on something you are or something you do, so you do not need to remember anything neither to hold any token.

Authentication methods by means of biometrics are a particular portion of security systems, with a good number of advantages over classical methods. However, there are also drawbacks (see Table 1).

Depending on the application, one of the previous methods, or a combination of them, will be the most appropriate. This article describes the main issues to be known for decision making, when trying to adopt a biometric security technology solution.

MAIN FOCUS OF THE ARTICLE

This article presents an overview of the main topics related to biometric security technology, with the central purpose to provide a primer on this subject.

Biometrics can offer greater security and convenience than traditional methods for people recognition. Even if we do not want to replace a classic method

Table 1. Advantages and drawbacks of the three main authentication method approaches

Authentication method	Advantages	Drawbacks
Handheld tokens (card, ID, passport, etc.)	▪ A new one can be issued. ▪ It is quite standard, although moving to a different country, facility, etc.	▪ It can be stolen. ▪ A fake one can be issued. ▪ It can be shared. ▪ One person can be registered with different identities.
Knowledge based (password, PIN, etc.)	▪ It is a simple and economical method. ▪ If there are problems, it can be replaced by a new one quite easily.	▪ It can be guessed or cracked. ▪ Good passwords are difficult to remember. ▪ It can be shared. ▪ One person can be registered with different identities.
Biometrics	▪ It cannot be lost, forgotten, guessed, stolen, shared, etc. ▪ It is quite easy to check if one person has several identities. ▪ It can provide a greater degree of security than the other ones.	▪ In some cases a fake one can be issued. ▪ It is neither replaceable nor secret. ▪ If a person's biometric data is stolen, it is not possible to replace it.

(**password** or handheld token) by a biometric one, for sure, we are potential users of these systems, which will even be mandatory for new passport models. For this reason, it is useful to be familiarized with the possibilities of biometric security technology.

BIOMETRIC TRAITS

The first question is: Which characteristic can be used for biometric recognition? As common sense says, a good biometric trait must accomplish a set of properties. Mainly they are (Clarke, 1994), (Mansfield & Wayman, 2002):

- Universality: Every person should have the characteristic.
- Distinctiveness: Any two persons should be different enough to distinguish each other based on this characteristic.
- Permanence: the characteristic should be stable enough (with respect to the matching criterion) along time, different environment conditions, etc.
- Collectability: the characteristic should be acquirable and quantitatively measurable.
- Acceptability: people should be willing to accept the biometric system, and do not feel that it is annoying, invasive, etc.
- Performance: the identification accuracy and required time for a successful recognition must be reasonably good.
- Circumvention: the ability of fraudulent people and techniques to fool the biometric system should be negligible.

Biometric traits can be split into two main categories:

- Physiological biometrics: it is based on direct measurements of a part of the human body. Fingerprint (Maltoni et al., 2003), face, iris and hand-scan (Faundez-Zanuy, Navarro-Mérida, 2005) recognition belong to this group.
- Behavioral biometrics: it is based on measurements and data derived from an action performed by the user, and thus indirectly measures some characteristics of the human body. Signature

(Faundez-Zanuy, 2005c) , gait, gesture and key stroking recognition belong to this group.

However, this classification is quite artificial. For instance, the speech signal (Faundez-Zanuy and Monte, 2005) depends on behavioral traits such as semantics, diction, pronunciation, idiosyncrasy, etc. (related to socio-economic status, education, place of birth, etc.) (Furui, 1989). However, it also depends on the speaker's physiology, such as the shape of the vocal tract. On the other hand, physiological traits are also influenced by user behavior, such as the manner in which a user presents a finger, looks at a camera, etc.

Verification and Identification

Biometric systems can be operated in two modes, named identification and verification. We will refer to recognition for the general case, when we do not want to differentiate between them. However, some authors consider recognition and identification synonymous.

- Identification: In this approach no identity is claimed from the user. The automatic system must determine who the user is. If he/ she belongs to a predefined set of known users, it is referred to as closed-set identification. However, for sure the set of users known (learnt) by the system is much smaller than the potential number of people that can attempt to enter. The more general situation where the system has to manage with users that perhaps are not modeled inside the database is referred to as open-set identification. Adding a "none-of-the-above" option to closed-set identification gives open-set identification. The system performance can be evaluated using an identification rate.
- Verification: In this approach the goal of the system is to determine whether the person is the one that claims to be. This implies that the user must provide an identity and the system just accepts or rejects the users according to a successful or unsuccessful verification. Sometimes this operation mode is named authentication or detection. The system performance can be evaluated using the False Acceptance Rate (FAR, those situations where an impostor is accepted) and the False Rejection Rate (FRR, those situations where a user is incorrectly rejected), also known in detection

Figure 1. On the left: example of a DET plot for a user verification system (dotted line). The Equal Error Rate (EER) line shows the situation where False Alarm equals Miss Probability (balanced performance). Of course one of both errors rates can be more important (high security application versus those where we do not want to annoy the user with a high rejection/ miss rate). If the system curve is moved towards the origin, smaller error rates are achieved (better performance). If the decision threshold is reduced, higher False Acceptance/Alarm rates are achieved. On the right: Example of a ROC plot for a user verification system (dotted line). The Equal Error Rate (EER) line shows the situation where False Alarm equals Miss Probability (balanced performance).
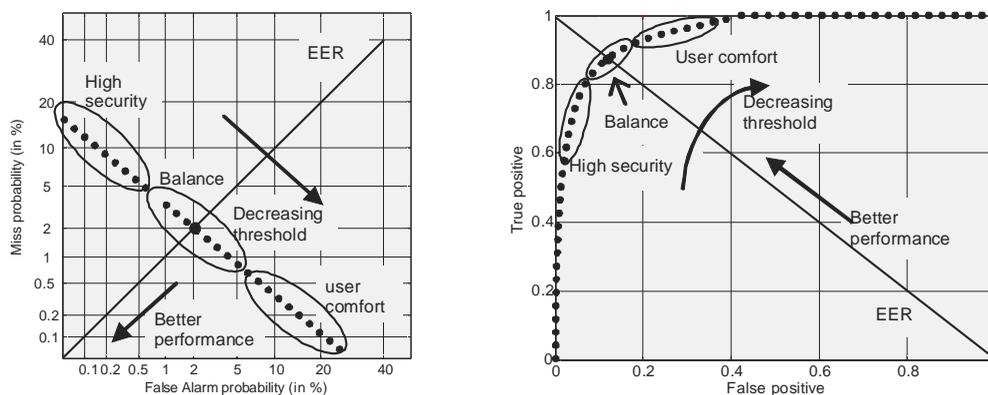
theory as False Alarm and Miss, respectively. There is a trade-off between both errors, which has to be usually established by adjusting a decision threshold. The performance can be plotted in a **ROC** (Receiver Operator Characteristic) or in a **DET** (Detection error trade-off) plot (Martin et al., 1989). DET curve gives uniform treatment to both types of error, and uses a logarithmic scale for both axes, which spreads out the plot and better distinguishes different well performing systems and usually produces plots that are close to linear. Note also that the ROC curve has symmetry with respect to the DET, i.e. plots the hit rate instead of the miss probability. DET plot uses a logarithmic scale that expands the extreme parts of the curve, which are the parts that give the most information about the system performance. Figure 1, on the left shows an example of DET of plot, and on the right shows a classical ROC plot.

For systems working in verification mode, the evolution of FAR and FRR versus the threshold setting is an interesting plot. Using a high threshold, no impostor can fool the system, but a lot of genuine users will be rejected. Contrarily, using a low threshold, there would

not be inconveniences for the genuine users, but it will be reasonably easy for a hacker to crack the system. According to security requirements, one of both taxes will be more important than the other one.

Is Identification Mode More Appropriate Than Verification Mode?

Certain applications lend themselves to verification, such as PC and network security, where, for instance, you replace your **password** by your fingerprint, but you still use your login. However, in forensic applications it is mandatory to use identification, because, for instance, latent prints lifted from crime scenes never provide their "claimed identity".

In some cases, such as room access (Faundez-Zanuy, 2004c), (Faundez-Zanuy & Fabregas 2005), it can be more convenient for the user to operate on identification mode. However, verification systems are faster because they just require one-to-one comparison (identification requires one to N, where N is the number of users in the database). In addition, verification systems also provide higher accuracies. For instance, a hacker has almost N times (Maltoni & al., 2003) more chance to fool an

Figure 2. General scheme of a biometric recognition system
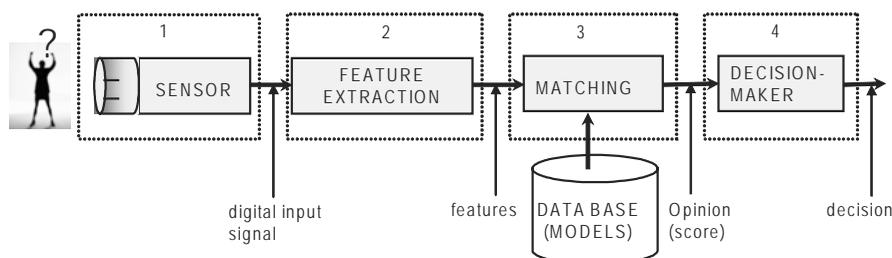
identification system than a verification one, because in identification he/she just needs to match one of the N genuine users. For this reason, commercial applications operating on identification mode are restricted to small-scale (at most, a few hundred users). Forensic systems (Faundez-Zanuy, 2005a), (Faundez-Zanuy, 2005b) operate in a different mode, because they provide a list of candidates, and a human supervisor checks the automatic result provided by the machine. This is related to the following classification, which is also associated to the application.

BIOMETRIC TECHNOLOGIES

Several biometric traits have been proven useful for biometric recognition. Nevertheless, the general scheme of a biometric recognition system is similar, in all the cases, to that shown in figure 2.

The scheme shown in figure 2 is also interesting for vulnerability study (Faundez-Zanuy, 2004b) and improvements by means of data fusion (Faundez-Zanuy, 2004d) analysis. In this paper, we will restrict to block number one, the other ones being related to signal processing and pattern recognition. Although common sense points out that good acquisition is enough for performing good recognition, at least for humans, this is not true. It must be taken into account that next blocks, numbered 2 to 4 in figure 2, are indeed fundamental. A good image or audio recording is not enough. Even for human beings, a rare disorder named **agnosia** exists. Those individuals suffering agnosia are unable to recognize and identify objects or persons despite having knowledge of the characteristics of the objects or persons. People with agnosia may have difficulty recognizing the geometric features of an object or face

or may be able to perceive the geometric features but do not know what the object is used for or whether a face is familiar or not. Agnosia can be limited to one sensory modality such as vision or hearing. A particular case is named face blindness or prosopagnosia (http://www.faceblind.org/research). Prosopagnosics often have difficulty recognizing family members, close friends, and even themselves. Agnosia can result from strokes, dementia, or other neurological disorders. More information about agnosia can be obtained from the National Organization for Rare Disorders (NORD http://www.rarediseases.org).

Table 2 summarizes some possibilities for different biometric traits acquisition. Obviously, properly speaking, some sensors require a digitizer connected at its output, which is beyond the scope of this paper. We will consider that block number one produces a digital signal which can be processed by a Digital Signal Processor (DSP) or Personal Computer (PC).

Figure 3 shows some biometric traits and their corresponding biometric scanners.

Security and Privacy

A nice property of biometric security systems is that security level is almost equal for all users in a system. This is not true for other security technologies. For instance, in an access control based on **password**, a hacker just needs to break only one password among those of all employees to gain access. In this case, a weak password compromises the overall security of every system that user has access to. Thus, the entire system's security is only as good as the weakest password (Prabhakar, Pankanti & Jain, 2003). This is especially important because good passwords are nonsense combinations of characters and letters, which

Table 2. Some biometric traits and possible acquisition sensors

Biometric trait	Sensor	Comments
Fingerprint	Ink+ paper + scanner	Classical method is becoming old-fashion, because the ink is annoying. However, it can acquire from nail to nail borders, and the other methods provide a limited portion of the fingerprint.
	Optical	It is the most widely used and easy-to-operate technology. It can acquire larger surfaces than the capacitive ones.
	Capacitive	They are easy to integrate into small, low-power and, low-cost devices. However, they are more difficult to operate than the optical ones (wet and/or warm fingers).
	Ultrasound	They are not ready for mass-market applications yet. However, they are more capable of penetrating dirt than the other ones, and are not subject to some of the image-dissolution problems found in larger optical devices.
Face	Photo-camera	Nowadays almost all the mobile phones have a photo-camera, enabling face recognition applications.
	Video-camera	A sequence of images alleviates some problems, such as face detection and offers more possibilities.
Speech	Microphone	The telephone system provides a ubiquitous network of sensors for acquiring speech signals.
Iris	Kiosk-based systems	The camera searches for eye position. They are the most expensive ones and the easier to operate.
	Physical access devices	The device requires some user effort: a camera is mounted behind a mirror. The user must locate the image of his eye within a 1-inch by 1-inch square surface on the mirror.
	Desktop cameras	The user must look into a hole and look at a ring illuminated inside.
Retina	Retina-scanner	A relative large and specialized device is required. It must be specifically designed for retina imaging. Image acquisition is not a trivial matter.
Signature	Ball pen + paper + scanner/camera	The system recognizes the signature analyzing its shape. This kind of recognition is known as "off-line", while the other ones are "on-line".
	Graphics tablet	It acquires the signature in real time. Some devices can acquire: position in x and y-axis, pressure applied by the pen, azimuth and altitude angles of the pen with respect to the tablet.
	PDA	Stylus operated PDAs are also possible. They are becoming more popular, so there are some potential applications.
Hand-geometry	Hand-scanning device	Commercial devices consist of a covered metal surface, with some pegs for ensuring the correct hand position. A series of cameras acquire three dimensional (3D) images of the back and the sides of the hand.
	Conventional scanner	Some research groups at universities have developed systems based on images acquired by a conventional document scanner. Thus, the cost is reduced, but the acquisition time is at least 15 seconds per image.
	Conventional camera	Some research groups at universities have developed systems based on images acquired by conventional cameras. Using some "bricolage" it is possible to obtain a 3D image.
Palm-print	Document scanner	Although there are not commercial applications, some research groups at universities have developed systems based on images acquired by a conventional document scanner.
Keystroke	keyboard	Although not used habitually, standard keyboards can measure how long keys are held down and duration between key instances, which is enough for recognition.

B

Figure 3. Examples of biometric traits and an examples of commercial scanners

Face acquired with digital camera	*2D hand geometry acquired with a document scanner and a 3D hand-geometry scanner*
Fingerprint acquired with optical sensor	*Iris and Iris desktop camera*
Speech *acquired with a microphone*	*Signature* *acquired with a graphics tablet*
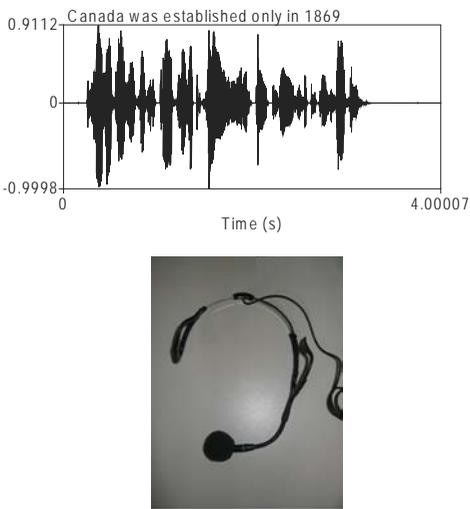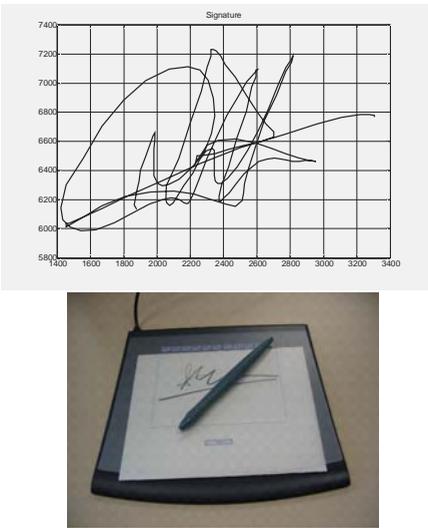
are difficult to remember (for instance, "Jh2pz6R+"). Unfortunately, some users still use passwords such as "password", "Homer Simpson" or their own name.

FUTURE TRENDS

Although biometrics offers a good set of advantages, it has not been massively adopted yet (Faundez-Zanuy, 2005d). One of its main drawbacks is that biometric data is not secret and cannot be replaced after being compromised by a third party. For those applications with a human supervisor (such as border entrance control), this can be a minor problem, because the operator can check if the presented biometric trait is original or fake. However, for remote applications such as internet, some kind of liveliness detection and **anti-replay attack** mechanisms should be provided. This is an emerging research topic. As a general rule, concerning security matters, a constant update is necessary in order to keep on being protected. A suitable system for the present time can become obsolete if it is not periodically improved. For this reason, nobody can claim that he/ she has a perfect security system, and even less that it will last forever.

Another interesting topic is privacy, which is beyond the scope of this article. It has been recently discussed in (Faundez-Zanuy, 2005a).

REFERENCES

Clarke R. (1994) "Human identification in information systems: management challenges and public information issues". December. Available in http://www.anu.edu.au/people/Roger.Clarke/DV/HumanID.html

Faundez-Zanuy M. (2004a) "Are Inkless fingerprint sensors suitable for mobile use? *IEEE Aerospace and Electronic Systems Magazine*, pp.17-21, April.

Faundez-Zanuy M. (2004b) "On the vulnerability of biometric security systems" *IEEE Aerospace and Electronic Systems Magazine* Vol.19 nº 6, pp.3-8, June.

Faundez-Zanuy M. (2004c) "Door-opening system using a low-cost fingerprint scanner and a PC". *IEEE Aerospace and Electronic Systems Magazine*. Vol. 19 nº 8, pp.23-26. August.

Faundez-Zanuy M. (2004d)"Data fusion in biometrics". *IEEE Aerospace and Electronic Systems Magazine*. Vol. 20 nº 1, pp.34-38, January.

Faundez-Zanuy M. (2005a) "Privacy issues on biometric systems". *IEEE Aerospace and Electronic Systems Magazine*. Vol. 20 nº 2, pp13-15, February.

Faundez-Zanuy M. (2005b)"Technological evaluation of two AFIS systems" *IEEE Aerospace and Electronic Systems Magazine*. Vol.20 nº 4, pp13-17, April.

Faundez-Zanuy M., Monte E. (2005) "State-of-the-art in speaker recognition". *IEEE Aerospace and Electronic Systems Magazine*. Vol.20 nº 5, pp 7-12 May.

Faundez-Zanuy M., Fabregas J, (2005)"Testing report of a fingerprint-based door-opening system". *IEEE Aerospace and Electronic Systems Magazine*. June.

Faundez-Zanuy M. (2005c) "State-of-the-art in signature recognition". *IEEE Aerospace and Electronic Systems Magazine*. July.

Faundez-Zanuy M. (2005d) "Biometric recognition: why not massively adopted yet?".*IEEE Aerospace and Electronic Systems Magazine*. Vol.20 nº 8, pp.25-28, August 2005

Faundez-Zanuy M., Navarro-Mérida G. M. (2005) "Biometric identification by means of hand geometry and a neural net classifier" *Lecture Notes in Computer Science* LNCS 3512, pp. 1172-1179, IWANN'05 June

Furui S. (1989) *Digital Speech Processing, synthesis, and recognition*., Marcel Dekker.

Li Stan Z. , Jain A. K. (2003) *Handbook of Face Recognition* Ed. Springer

Maltoni D., Maio D., Jain A. K., Prabhakar S. (2003) *Handbook of Fingerprint Recognition* Springer professional computing. 2003

Mansfield A. J., Wayman J. L., (2002) "Best Practices in Testing and Reporting Performance of Biometric Devices". Version 2.01. National Physical Laboratory Report CMSC 14/02. August.

Martin A., Doddington G., Kamm T., Ordowski M., and Przybocki M., (1997) "The DET curve in assessment of detection performance", V. 4, pp.1895-1898, *European speech Processing Conference Eurospeech 1997*

B

Prabhakar S., Pankanti S., Jain A. K. (2003) "Biometric recognition: security and privacy concerns" *IEEE Security and Privacy*, pp. 33-42, March/April

KEY TERMS

Automatic Identification: The system must determine who the user is.

Automatic Verification: The system must determine whether the person is the one that claims to be.

Behavioral Biometrics: Based on measurements and data derived from an action performed by the user, and thus indirectly measures some characteristics of the human body. Signature, gait, gesture and key stroking recognition belong to this group.

Equal Error Rate: System performance when False Acceptance Rate is identical to False Rejection Rate.

False Acceptance Rate: Ratio of impostors whose access is incorrectly permitted.

False Rejection Rate: Ratio of genuine users whose access is incorrectly denied.

Physiological Biometrics: Based on direct measurements of a part of the human body. Fingerprint, face, iris and hand-scan recognition belong to this group.

Blind Source Separation by ICA

Miguel A. Ferrer
University of Las Palmas de Gran Canaria, Spain

Aday Tejera Santana
University of Las Palmas de Gran Canaria, Spain

INTRODUCTION

This work presents a brief introduction to the blind source separation using independent component analysis (ICA) techniques. The main objective of the blind source separation (BSS) is to obtain, from observations composed by different mixed signals, those different signals that compose them. This objective can be reached using two different techniques, the spatial and the statistical one. The first one is based on a microphone array and depends on the position and separation of them. It also uses the directions of arrival (DOA) from the different audio signals.

On the other hand, the statistical separation supposes that the signals are statistically independent, that they are mixed in a linear way and that it is possible to get the mixtures with the right sensors (Hyvärinen, Karhunen & Oja, 2001) (Parra, 2002).

The last technique is the one that is going to be studied in this work. It is due to this technique is the newest and is in a continuous development. It is used in different fields such as natural language processing (Murata, Ikeda & Ziehe, 2001) (Saruwatari, Kawamura & Shikano, 2001), bioinformatics, image processing (Cichocki & Amari, 2002) and in different real life applications such as mobile communications (Saruwatari, Sawai, Lee, Kawamura, Sakata & Shikano, 2003).

Specifically, the technique that is going to be used is the Independent Component Analysis (ICA). ICA comes from an old technique called PCA (Principal Component Analysis) (Hyvärinen, Karhunen & Oja, 2001) (Smith, 2006). PCA is used in a wide range of scopes such as face recognition or image compression, being a very common technique to find patterns in high dimension data.

The BSS problem can be of two different ways; the first one is when the mixtures are linear. It means that the data are mixed without echoes or reverberations, while the second one, due to these conditions, the mixtures are convolutive and they are not totally independent because of the signal propagation through dynamic environments. It is the "Cocktail party problem". Depending on the mixtures, there are several methods to solve the BSS problem. The first case can be seen as a simplification of the second one.

The blind source separation based on ICA is also divided into three groups; the first one are those methods that works in the time domain, the second are those who works in the frequency domain and the last group are those methods that combine frequency and time domain methods. A revision of the technique state of these methods is proposed in this work.

BACKGROUND

The problem consists in several sources that are mixed in a system, these mixtures are recorded and then they have to be separated to obtain the estimations of the original sources. As was mentioned above, BSS problems can be of two different types; the first one, when the mixtures are linear, see equation 3, and the second one, when the mixtures are convolutive, see equation 5.

In the first case each source signal is multiplied by a constant which depends on the environment, and then they are added. Convolutive mixtures are not totally independent due to the signal propagation through dynamic environments. This makes that the signals are not simply added. The first case is the ideal one, and the second is the most common case, because in real room recordings the mixing systems are of this type. The following figure shows the mixing system in the case of two sources two mixtures:

Where X_1 and X_2 are the independent signals, Y_1 and Y_2 are the mixing of the different X_j, and $\overline{H}$ is the mixing system that can be seen in a general form as:

Figure 1. 2 sources – 2 mixtures system.
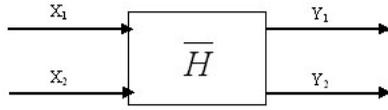
$$\overline{H} = \begin{bmatrix} h_1 & ... & h_{1j} \\ ... & ... & ... \\ h_{i1} & ... & h_j \end{bmatrix}$$

(1)

The h_{ij} are FIR filters, each one represents an acoustic transference multipath function from source, i, to sensor, j. i and j represent the number of sources and sensors. Now it is necessary to remember the first condition that makes possible the blind source separation:

"The number of sensors must be greater than or equal to the number of sources."

Taking this into account, the problem for two sensors, in a general form, can be represented as:

$$Y_1 = X_1 * h_1 + X_2 * h_2$$ (2.1)

$$Y_2 = X_1 * h_2 + X_2 * h_2$$ (2.2)

Generally, there are n source signals statistically independent $X(t) = [X_1(t),...,X_n(t)]$, and m observed mixtures that are linear and instantaneous combinations of the previous signals $Y(t) = [Y_1(t),...,Y_n(t)]$. Beginning with the linear case, the simplest case, the mixtures are:

$$Y_i(t) = \sum_{j=1}^{n} h_j \cdot X_j(t)$$ (3)

Now, we need to recover X(t) from Y(t). It is necessary to estimate the inverse matrix of H, where h_{ij} are contained. Once we have this matrix:

$$\hat{X}(t) = \overline{W} \cdot Y(t)$$ (4)

where $\hat{X}(t)$ contains the estimations of the original source signals, and $\overline{W}$ is the inverse mixing matrix. Now we have defined the simplest case, it is time to explain the general case that involves convolutive mixtures.

The whole process, which includes mixing and separation process, and that has been described before for linear mixtures, is defined as in Figure 2.

The process will be the following; first a set of source signals (X) pass through an unknown system $\overline{H}$. The output (Y) contains all the mixtures. Y is equalized with an inverse estimated system $\overline{W}$, which has to give an estimation of the original source signals ($\hat{X}$).

Given access to N sensors with a number of sources less than or equal to N, all with unknown direct and cross channels, the objective is to recover all the unknown sources. Here arises the second condition to obtain the source separation:

"In blind source separation using ICA, it is assumed that we only know the probability density functions of the non-Gaussian and independent sources."

So we have to obtain $\overline{W}$, and it must be that:

$$\hat{X} = \overline{W} * Y$$ (5)

Here, as it was mentioned above, $\hat{X}$ are the estimations of the original source signals, Y are the observations, and $\overline{W}$ is the inverse mixing filter.

BLIND SOURCE SEPARATION BY ICA

This article presents different methods to solve the blind source separation, more exactly those that are based on independent component analysis (ICA). First, methods for the linear mixtures are going to be described, and then we are going to divide the methods for convolutive mixtures in three groups depending on the domain; frequency domain, time domain or both.

Blind Source Separation for Linear Mixtures

The blind source separation for linear mixtures is a particular case of the convolutive one. So methods

Figure 2. BSS general problem
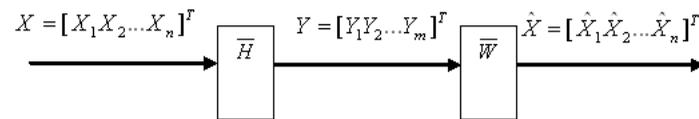
designed for convolutive mixtures must solve the problem of the linear mixtures in theory. In this case, we have decided to describe some linear methods separately due to there being some important methods specialized in this case.

There are different methods such as Infomax, which is based on the maximization of the information (Bell & Sejnowski, 1995), the one based on minimization of the mutual information (Hyvärinen, Karhunen & Oja, 2001), or methods which use tensors. The methods that are going to be described in this paper are FastICA and JADE, they have been selected due to they are two of the most famous methods for the linear case of BSS, and because they are a right first step into the BSS.

The first one, and maybe the most known, is FastICA (Hyvärinen, Karhunen & Oja, 2001). The FastICA algorithm is a computationally highly efficient method for performing the estimation of ICA. It uses a fixed-point iteration scheme that has been found in independent experiments to be 10-100 times faster than conventional gradient descent methods for ICA. Another advantage of the FastICA algorithm is that it can be used to perform projection pursuit as well, thus providing a general-purpose data analysis method that can be used both in an exploratory fashion and for estimation of independent components (or sources) (Hyvärinen, Karhunen & Oja, 2001) (Parra, 2002). This algorithm is available in a toolbox that is very easy to use (Helsinki University of Technology, 2006).

Another method that is very common in blind source separation of linear mixtures is JADE (Cardoso, 1993) (Hyvärinen, Karhunen & Oja, 2001). JADE (Join approximate diagonalization of eigenmatrices) refers to one principle of solving the problem of equal eigenvalues of the cumulant tensor. In this algorithm, the tensor EVD is considered more as a preprocessing step. A method closely related to JADE is given by the eigenvalue descomposition of the weighted correlation

matrix. For historical reasons, the basic method is simply called fourth-order blind identification (FOBI) (Hyvärinen, Karhunen & Oja, 2001).

Blind Source Separation for Convolutive Mixtures

Once the linear problem has been described, it is time to explain how to solve the problem with convolutive mixtures. This case is more complex than the linear one, as was presented in the Background. When the mixtures are convolutive the problem is also called blind deconvolution.

To solve this problem, several methods have been designed. They can be divided into three groups depending on the domain: time domain, frequency domain or both.

When the algorithm works in the frequency domain, the convolutive mixtures can be simplified into simultaneous ones by means of the frequency transform. This makes easier the convergence of the separation filter (Choi, Cichocki, Park & Lee, 2005). So these algorithms can increase the speed of convergence and reduce the computational load. But it has a cost; to maintain the computational efficiency these algorithms need to increase the length of the frame when the window frame increases. The data is reduced and can provoke insufficiency of the learning data. So the efficiency of the algorithm is degraded.

Some examples of these algorithms are: blind source separation in the wavelet domain, recursive method (Ding, Hikichi, Niitsuma, Hamatsu & Sugai, 2003), time delay decorrelation (Lee, Ziehe, Orglmeister & Sejnowski, 1998), ICA algorithm and beamforming (Saruwatari, Kawamura & Shikano, 2001) or the FIR matrix toolbox 5.0 (Lambert, 1996).

If the algorithm works in the time domain, we can work with wide band audio signals, where the assump-

tion of independence is kept. The disadvantages are that they produce a high computational load when they work with huge separation matrixes, and the convergence is slow, overall when the signals are voices.

Some algorithms that work in time domain are SIMO – ICA (Saruwatari, Sawai, Lee, Kawamura, Sakata & Shikano, 2003), filter banks (Choi, Cichocki, Park & Lee, 2005) or time-domain fast fixed-point algorithms for convolutive ICA (Thomas, Deville & Hosseini, 2006).

But we can also combine the two previous methods to compensate the advantages and disadvantages of each other, for example in Multistage ICA (Nishikawa, Saruwatari, Shikano, Araki & Makino, 2003); FDICA (frequency domain ICA) and TDICA (time domain ICA) are combined with the objective of attaining better efficiency that is possible. This algorithm has a stable behaviour, but the computational load is high.

FUTURE TRENDS

Blind source separation has wide scope that is in continuous development and several authors are working on the design or modification of different methods. In this work, different algorithms have been presented to solve the blind source separation problem.

Future trends will be the design of on line methods that allow the implementation of these algorithms in real life applications such as voice recognition or free hand devices. The algorithms can also be improved with the aim of reaching more efficient and accurate separation of the mixtures, linear or convolutive. It will help in different systems as a first step for identifying speakers, for example in conferences, videoconferences or similar.

CONCLUSION

This article shows different ways to solve blind source separation. It also illustrates that the problem can be of two forms depending on the mixtures. If we have linear mixtures the system will have a different behaviour that if we have convolutive ones.

As has been described above, the methods can be also divided into two types, the frequency domain and the time domain algorithms. The first type has a faster convergence than the time domain type, but

it can work incorrectly if data are insufficient. Time domain methods have more stable behaviour than the frequency algorithms, but a higher computational load. To trade off the advantages with the disadvantages of each type of method, multistage algorithms have been proposed.

REFERENCES

Cardoso, JF. (1999) High-Order Contrasts for Independent Component Analysis. Neural Computation, 11, 157-192.

Choi, S., Cichocki, A., Park, HM., Lee, SY. (2005). Blind Source Separation and Independent Component Analysis: A review. Neural Information Processing – Letters and Reviews. Volume 6, number 1.

Cichocki, A., Amari, S. I., (2002). Adaptive Blind Signal and Image Processing. John Wiley & Sons.

Ding, S., Hikichi, T., Niitsuma, T., Hamatsu, M., Sugai, K. (2003). Recursive method for Blind Source Separation and its applications to real-time separations of acoustic signals. Research group, Advanced Technology Development Department, R & D Division, Clarion Co., LTD.

Helsinki University of Technology, Laboratory of Computer and Information Science. http://www.cis.hut.fi/projects/ica/fastica/ , (last update 2006, last visit 10/04/07).

Hyvärinen, A., Karhunen, J., Oja, E. (2001). Independent Component Analysis. John Wiley & Sons.

Lambert, R.H. (1996). Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixture. Thesis, University of Sourthern California, Department of Electrical Engineering.

Lee, TW., Ziehe, A., Orglmeister, R., Sejnowski, T. (1998). Combining Time-Delayed Decorrelation and ICA: towards solving the Cocktail Party Problem. In proceedings of International Conference on Acoustics, Speech and Signal Processing. Volume 2, 1249 – 1252.

Murata, N., Ikeda, S., Ziehe, A. (2001). An approach to blind source separation based on temporal structure of speech signals. Neurocomputational, volume 41, 1 – 24.

B

Nishikawa, T., Saruwatari, H., Shikano, K., Araki, S., Makino, S. (2003). Multistage ICA for blind source separation of real acoustic convolutive mixture. In proceedings of International Conference on ICA and BSS. 523 – 528.

Parra, L. (2002). Tutorial on Blind Source Separation and Independent Component Analysis. Adaptive Image & Signal Processing Group, Sarnoff Corporation.

Saruwatari, H., Sawai, K., Lee A., Kawamura T., Sakata M., Shikano, K. (2003). Speech enhancement and recognition in car environment using Blind Source Separation and subband elimination processing. In proceedings International Workshop on Independent Component Analysis and Signal Separation, 367 – 372.

Saruwatari, H., Kawamura T., Shikano, K. (2001). Blind Source Separation for speech based on fast convergence algorithm with ICA and beamforming. In oroceedings EUROSPEECH2001, 2603 – 2606.

Smith, L. I. (2006). A tutorial on Principal Component Analysis. http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

Thomas, J., Deville, Y., Hosseini, S. (2006). Time-domain fast fixed-point algorithms for convolutive ICA. Signal processing letters, IEEE. Volume 13, Issue 4, 228 – 231.

Bell A.J., Sejnowski T.J. (1995). An information maximization approach to blind separation and blind deconvolution, Neural Computation, 7, 6, 1129 – 1159.

KEY TERMS

Blind Source Separation: The problem of separating from mixtures, the source signals that compose them.

Cocktail Party Problem: A particular case of blind source separation where the mixtures are convolutive.

Convolutive Mixtures: Mixtures that are not linear due to echoes and reverberations, they are not totally independent because of the signal propagation through dynamic environments.

HOS (High Order Statistics): Higher order statistics is a field of statistical signal processing that uses more information than autocorrelation functions and spectrum. It uses moments, cumulants and polyspectra. They can be used to get better estimates of parameters in noisy situations, or to detect nonlinearities in the signals.

ICA (Independent Component Analysis): Techniques based on statistical concepts such as high order statistics.

Linear mixtures: Mixtures that are linear combinations of the different sources that compose them.

Statistically Independent: Given two events, the occurrence of one event makes it neither more nor less probable that the other occurs.

Chaotic Neural Networks

Emilio Del-Moral-Hernandez
University of São Paulo, Brazil

INTRODUCTION

Artificial Neural Networks have proven, along the last four decades, to be an important tool for modelling of the functional structures of the nervous system, as well as for the modelling of non-linear and adaptive systems in general, both biological and non biological (Haykin, 1999). They also became a powerful biologically inspired general computing framework, particularly important for solving non-linear problems with reduced formalization and structure. At the same time, methods from the area of complex systems and non-linear dynamics have shown to be useful in the understanding of phenomena in brain activity and nervous system activity in general (Freeman, 1992; Kelso, 1995). Joining these two areas, the development of artificial neural networks employing rich dynamics is a growing subject in both arenas, theory and practice. In particular, model neurons with rich bifurcation and chaotic dynamics have been developed in recent decades, for the modelling of complex phenomena in biology as well as for the application in neuro-like computing. Some models that deserve attention in this context are those developed by Kazuyuki Aihara (1990), Nagumo and Sato (1972), Walter Freeman (1992), K. Kaneko (2001), and Nabil Farhat (1994), among others. The following topics develop the subject of Chaotic Neural Networks, presenting several of the important models of this class and briefly discussing associated tools of analysis and typical target applications.

BACKGROUND

Artificial Neural Networks (ANNs) is one of the important frameworks for biologically inspired computing. A central characteristic in this paradigm is the desire to bring to computing models some of the interesting properties of the nervous system such as adaptation, robustness, non-linearity, and the learning through examples.

When we focus on biology (real neural networks), we see that the signals generated in real neurons are used in different ways by the nervous system to code information, according to the context and the functionality (Freeman, 1992). Because of that, in ANNs we have distinct model neurons, such as models with graded activity based on frequency coding, models with binary outputs, and spiking models (or pulsed models), among others, each one giving emphasis to different aspects of neural coding and neural processing. Under this scenario, the role of neurodynamics is one of the target aspects in neural modelling and neuro-inspired computing; some model neurons include aspects of neurodynamics, which are mathematically represented through differential equations in continuous time, or difference equations in discrete time. As described in the following topic, dynamic phenomena happen at several levels in neural activity and neural assembly activity (in internal neural structures, in simple networks of interacting neurons, and in large populations of neurons). The model neurons particularly important for our discussion are those that emphasize the relationship between neurocomputing and non-linear dynamical systems with bifurcation and rich dynamic behaviour, including chaotic dynamics.

NEUROCOMPUTING AND THE ROLE OF RICH DYNAMICS

The presence of dynamics in neural functionality happens even at the more detailed cellular level: the well known Hodgkin and Huxley model for the generation and propagation of action potentials in the active membrane of real neurons is an example; time dependent processes related to synaptic activity and the post synaptic signals is another example. Dynamics also appears when we consider the oscillatory behaviour in real neurons under consistent stimulation. Additionally, when we consider neural assemblies, we also observe the emergence of important global dynamic behaviour for the production of complex functions.

As discussed ahead, non-linearity is an essential ingredient for complex functionality and for complex dynamics; there is a clear contrast between linear dynamic systems and non-linear dynamic systems, in what respect their potential for the production of rich and diverse behaviour.

Role of Non-Linear Dynamics in the Production of Rich Behaviour

In linear dynamical systems, both in continuous time and in discrete time, the autonomous dynamical behaviour is completely characterized through the system's natural modes, either the harmonic oscillatory modes, or the exponentially decaying modes (in the theory of linear dynamical systems, these are represented by frequencies and complex frequencies). The possible dynamic outcomes in linear systems are thus limited to the universe of linear combinations of these natural modes. These modes can have their properties of amplitudes and frequencies controlled through parameters of the system, but not their central properties such as the nature of the produced waveforms. Since the number of natural modes of linear systems is closely related to the number of state variables, we have that small networks (of linear dynamic elements) can produce only limited diversity of dynamical behaviour.

The scenario becomes completely different in non-linear systems. Non-linearity promotes rich dynamic behaviour, obtained by changing the stability and instability of different attractors. These changes give place to bifurcation phenomena (transitions between dynamic modalities with distinct characteristics) and therefore to diversity of dynamic behaviour. In non-linear systems, we can have a large diversity of dynamical behaviours, with the potential production of infinite number of distinct waveforms (or time series, for discrete time systems). This can happen for systems with very reduced number of state variables: just three in continuous time, or just one state variable in discrete time, are enough to allow bifurcation among different attractors and potential cascades of infinite bifurcations leading to chaos. In our context, this means obtaining rich attractor behaviour even from very simple neural networks (i.e., networks with a small number of neurons).

In summary, the operation of chaotic neural networks explores the concepts of attractors, repellers, limit cycles, and stability (see the topic Terms and Definitions for details on these concepts) of trajectories in the multidimensional state space of the neural network, and more specifically, the dense production of destabilization of cyclic trajectories with cascading to chaotic behaviour. This scenario allows for the blend of ordered behaviour and chaotic dynamics, and the presence of fractal structure and self-similarity in the rich landscape of dynamic attractors.

MODEL NEURONS WITH RICH DYNAMICS, BIFURCATION AND CHAOS

We can look at chaotic elements that compose neuro-like architectures from several different perspectives. They can be looked at as emergent units with rich dynamics that are produced by the interaction of classical model neurons, such as the sigmoidal model neurons based on frequency coding (Haykin, 1999), or the integrate and fire spiking model neurons (Farhat, 1994). They can also correspond to the modelling of dynamical behaviour of neural assemblies, approached as a unity (Freeman, 1992). Finally, they can be tools for approximate representation of aspects of complex dynamics in the nervous system, paying attention mainly to the richness of attractors and blend of ordered and erratic dynamics, and not exactly to the details of the biological dynamics (DelMoral, 2005; Kaneko, 2001). Ahead we describe briefly some of the relevant model neurons in the context of chaotic neural networks.

Aihara's Chaotic Model Neuron. One important work in the context of chaotic neural networks is the model neuron proposed by Kazuyuki Aihara and collaborators (1990). In it, we have self-feedback of the neuron's state variable, for representing the refractory period in real neurons. This makes possible rich bifurcation and cascading to chaos. His work extends previous models in which some elements of dynamics were already present. In particular, we have to mention the work by Caianiello (1961), in which the past inputs have impact on the value of the present state of the neuron, and the work by Nagumo and Sato (1972), which incorporates an exponential decay memory. Aihara's model included memory for the inputs of the model neuron as well as for its internal state. It also included continuous transfer functions, an essential ingredient for rich bifurcation, fractal structure and cascading to chaos. Equation 1 shows a simplified form

of this model: x_i is the node state, while the x_j regard neighboring nodes, t is discrete time, f a continuous function, k_f and k_r decay constants, and w_{ij} generic coupling strengths:

$$x_i(t+1) = f\left(\sum_{j=1}^{M} w_{ij} \sum_{d=0}^{t} k_f^d x_j(t-d) - \alpha \sum_{d=0}^{t} k_r^d x_i(t-d)\right)$$

(1)

Adachi's Associative Memory. Another proposal that can be mentioned here is that of Adachi, co-authored with Aihara. It uses Aihara's chaotic neuron for the implementation of associative memories, with coupling strengths among nodes, w_{ij}, given by Hebbian-like correlation measures (Adachi, 1997). Equation 2 defines w_{ij} for this model, in a memory storing M binary strings x^p.

$$w_{ij} = \sum_{p=1}^{M} (x_i^p - \bar{x})(x_j^p - \bar{x})$$

(2)

Nabil Farhat's Bifurcating Neuron. Nabil Farhat and collaborators introduced the "Bifurcation Neuron", in which the phenomena of bifurcation and cascade to chaotic dynamics emerge from a pulsed model of type "Integrate and Fire" (Farhat, 1994). This work and some of its following developments have direct relationship to a class of association and pattern recovery architectures developed by the author of this article: chaotic neural networks based on **Recursive Processing Elements,** or **RPEs** (DelMoral, 2005 ; DelMoral, 2007). RPEs are parametric recursions that are coupled through modulation of their bifurcation parameters (we say we have *parametric coupling*), for the formation of meaningful collective patterns. Node dynamics is mathematically defined through a first order parametric recursion:

$$x(t+1) = R_p(x(t)$$

(3a)

example: x(t + 1) = p. [x(t).(1-x(t))]

(3b)

This recursion R_p (parameterized by the "bifurcation parameter" p) links consecutive values of the state variable x, which evolves in discrete time t. It is interesting to comment that first order non-linear recursions are very simple mathematical systems with very rich dynamical behavior (Fig.1 shows an illustrative bifurcation diagram).

Kunihiko Kaneko's Coupled Map Lattices - CMLs. These structures, initially conceived for the modelling of spatio-temporal physical systems, employ the idea of chaotic recursive maps (similar to the RPEs above described) that interact through diffusive-like coupling (Kaneko, 2001). Equation 4 represents a CML: i identifies a node of a linear lattice, and ε the amount of coupling.

Figure 1. Bifurcation diagram for the logistic recursion, where Rp(x) = p.x.(1-x)
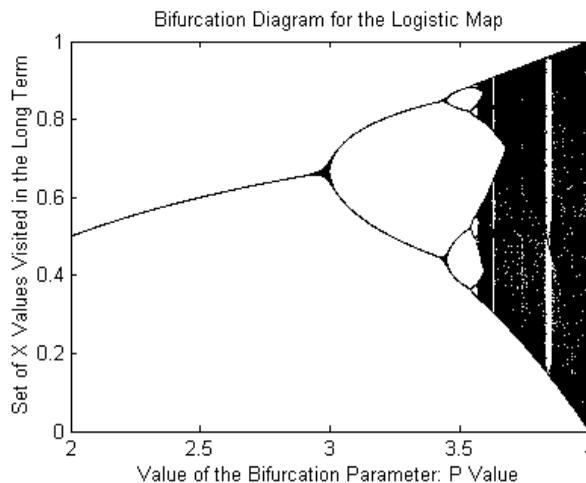
$$x_i(t+1) = (1-\varepsilon).R_p(x_i(t)) + \frac{\varepsilon}{2}.R_p(x_{i-1}(t)) + \frac{\varepsilon}{2}.R_p(x_{i+1}(t))$$

$$(4)$$

Non-Linear Dynamics Tools

The classical tools developed for the study of non-linear dynamic systems with bifurcation and diverse behaviour (Hilborn, 1994) are important elements in the study and characterization of chaotic neural networks (Kaneko, 2001 ; DelMoral, 2007). Among the important ones, we can mention the Bifurcation Diagrams (see Fig.1), which are useful for the representation of the long-term behavior in parametric non-linear dynamical systems, as well as for the representation of their bifurcations and their parameter ranges for ordered behavior and chaotic behavior. We can also mention the Lyapunov exponents, for the quantitative evaluation of sensibility to initial conditions, the Entropy measures, for the quantification of trajectory complexity, the Return Maps, for the characterization of recursive rules, and the Web Diagrams, for the illustration of attractor and repeller trajectories (Hilborn, 1994 ; Devaney, 1989 ; Kaneko, 2001).

COLLECTIVE DYNAMIC BEHAVIOUR, ATTRACTOR NETWORKS AND APPLICATION SCENARIOS

A more complex and richer scenario can be created through the coupling of several units having rich dynamic behavior at the single node level. The following paragraphs detail some of the emergent collective phenomena that appear in networks of coupled chaotic elements and are explored for information coding and processing:

- **Multidimensional attractors**. In the multidimensional attractor behavior, similarly to attractors at the single node level (see entry in the topic Terms and Definitions, at the end of this article), we have the evolution in time of the network state towards a limited repertoire of preferential collective trajectories, which emerge in the long-term. The concept of Multidimensional Attractors is central to the *Attractor Networks* paradigm: high dimen-

sional dynamical systems whose long-term states represent relevant information. In the particular case of chaotic neural networks, the evolution of the network state is usually composed by an initial chaotic phase and a gradual approximation to ordered limit cycles (Kaneko, 2001 ; DelMoral, 2005 ; Freeman, 1992).
- **Clustering of nodes' activities**. Here we have, due to coupling and network self organization, the formation of groups of nodes exhibiting activities which are identical or similar in some sense (Kaneko, 2001).
- **Synchronization of the nodes' cycling** (or phase locking). In this type of collective phenomena, which is a particular case of clustering, the cyclic activities of nodes belonging to a cluster have the same period, and they operate with constant relative phases (DelMoral, 2005 ; DelMoral, 2007).

With collective structures (multiple coupled neurons), complex functionalities can potentially be implemented, through the exploration of the multidimensional nature of the state variables: image understanding, processing of multiple sensory information, multidimensional logical reasoning, complex motor control, memory, association, hetero-association, decision making and pattern recognition (DelMoral, 2007). In networks of coupled elements with rich dynamics, the above collective phenomena are explored for the representation and processing of meaningful information. A stored memory, in association and pattern recovery tasks, or a class label, in pattern recognition tasks, for example, can be represented through the specific collective attractors of the network (DelMoral, 2005). Concretely, the representation of information can happen through different quantitative features of the relevant attractors. We can have the coding of analog information through the amplitude of oscillations of the state variables, or even through the sequences of values visited by the state variables in limit cycles. We can also have the coding of class labels through the periods of closed trajectories, through the phase of cycling of closed trajectories, or even through mixed forms involving several of these coding modalities. In addition, clustering and synchronization can be used for spatial segmentation of information.

Blend of Order and Complex Dynamics

A macroscopic phenomenon that relates to the global behavior of a coupled structure composed of several neurons with rich dynamics (as the models described in previous topics) is the interplay between ordered behavior and disordered behavior. In many circumstances, we can look at the network's state evolution as switching between situations of ordered behavior and situations of apparently erratic behavior. The blend of ordered and erratic behaviour is explored for the representation of meaningful information (order) and the rich search in the state space for stored patterns (chaotic search).

This blend of ordered and erratic behaviour appears in many different classes of model neurons and associated architectures, such as for example, the Bifurcation Neuron, K Sets structures, RPEs architectures and many others.

FUTURE TRENDS

Since chaotic neural networks are a relatively recent subject of research, there are many different directions in which the field can potentially progress. We will just mention some of these directions.

An important possibility that we can identify is the exploration of rich dynamics, fractal structure and diversity of dynamical behaviour, in the modelling and emulation of higher cognitive functions. We can mention for example that part of the current research on consciousness and cognition addresses the possible roles of complex dynamics on these high level functions (Perlovsky & Kozma, 2007).

We also see spiking model neurons, a fast growing research area, and neural oscillators as natural scenarios for the emergence of rich dynamic phenomena and as potential substrates for computing with rich dynamics (Gerstner, 2002). We add that there are several efforts for the implementation of spiking models in electronic form (DelMoral, 2003); these efforts can also have an important role in the context of brain-computer interfaces based on microelectrode arrays and associated electronics, another fast growing research area.

CONCLUSION

Chaotic Neural Networks and the associated chaotic model neurons discussed here are part of a current trend in neural modelling and artificial neurocomputing that moves the emphasis of artificial model neurons, from functional analysis and functional synthesis, particularly evident in neural architectures such as the MLPs, to a more balanced blend which involves also elements of neurodynamics and explores extensively the paradigm of attractor networks. With this ongoing move, the tendency is to have more powerful modelling tools for the study of the nervous system and more powerful elements for the development of neuro-like computing environments.

REFERENCES

Aihara, K., Takabe, T., & Toyoda, M. (1990). Chaotic neural networks. Physica Letters A, 144(6,7), 333-340.

Adachi, M., & Aihara, K. (1997). Associative dynamics in a chaotic neural network. Neural Networks, 10, 83-98.

Caianiello, E. R. (1961). Outline of a theory of thought-processes and thinking machines. Journal on Theoretical Biology, 2, 204-235.

Del-Moral-Hernandez, E., Gee-Hyuk Lee, & Farhat, N. (2003). Analog realization of arbitrary one dimensional maps. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 50, 1538-1547.

Del-Moral-Hernandez, E. (2005). Non-homogenous neural networks with chaotic recursive nodes: Connectivity and multi-assemblies structures in recursive processing elements architectures, Neural Networks, 18(5,6), 532-540.

Del-Moral-Hernandez, E. (2007). Recursive nodes with rich dynamics as modelling tools for cognitive functions. Chapter in L. Perlovsky & R. Kozma (Eds.), Neurodynamics of higher level cognition and consciousness. New York - Berlin - Heidelberg: Springer-Verlag.

Devaney, R. L. (1989). An introduction to chaotic dynamical systems (2nd ed.). Redwood City, CA: Addison-Wesley.

Farhat, N. H., S-Y Lin, & Eldelfrawy, M. (1994). Complexity and chaotic dynamics in spiking neuron embodiment. SPIE Critical Review, CR55, 77-88.

Freeman, W. J. (1992). Tutorial on neurobiology: From single neuron to brain chaos. International Journal of Bifurcation and Chaos, 2(3), 451-482.

Gerstner, W., Kistler, & W. M. (2002). Spiking neuron models: Single neurons, populations, plasticity. Cambridge, UK: Cambridge University Press.

Haykin, S. (1999). Neural networks: A comprehensive foundation (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Hilborn, R.C. (1994). Chaos and nonlinear dynamics: An introduction for scientists and engineers. New York: Oxford University Press.

Ishii, S. et al. (1993). Associative memory using spatiotemporal chaos. In International Joint Conference on Neural Networks 1993 (Nagoya), 3, 2638-2641.

Kaneko, K., & Tsuda, I. (2001) Complex systems: Chaos and beyond: A constructive approach with applications in life sciences. Springer-Verlag.

Kelso, J. A. S. (1995). Dynamic patterns - The self-organization of brain and behaviour. Cambridge, MA: The MIT Press.

Kozma, R., & Freeman, W. J. (2001). Control of Mesoscopic / Intermediate-Range Spatio-Temporal Chaos in the Cortex. In American Control Conference 2001, 1, 263-268.

Nagumo, J., & Sato, S. (1972). On a response characteristic of a mathematical neuron model. Kybernetic, 10, 155-164.

Perlovsky, L., & Kozma, R. (Eds.) (2007). Neurodynamics of higher level cognition and consciousness. New York - Berlin - Heidelberg: Springer-Verlag.

Siegelmann, H. T. (1995). Computation beyond the Turing limit. Science, 268, 545-548.

KEY TERMS

Artificial Neurons / Model Neurons: Mathematical description of the biological neuron, in what respects representation and processing of information. These models are the processing elements that compose an artificial neural network. In the context of chaotic neural networks, these models include the representation of aspects of complex neurodynamics.

Attractors, Repellers and Limit Cycles: These three concepts are **r**elated to the concept of dynamic modality and they regard the long-term behaviour of a dynamical system. Attractors are trajectories of the system state variable that emerge in the long-term, with relative independence with respect to the exact values of the initial conditions. These long-term trajectories can be either a point in the state space (a static asymptotic behaviour), named fixed-point, a cyclic pattern (named limit cycle), or even a chaotic trajectory. Repellers correspond, qualitatively speaking, to the opposite behaviour of attractors: given a fixed-point or a cyclic trajectory of a dynamic system, they are called repeller-type trajectories if small perturbations can make the system evolve to trajectories that are far from the original one.

Bifurcation and Diverse Dynamics: The concept of bifurcation, present in the context of non-linear dynamic systems and theory of chaos, refers to the transition between two dynamic modalities qualitatively distinct; both of them are exhibited by the same dynamic system, and the transition (bifurcation) is promoted by the change in value of a relevant numeric parameter of such system. Such parameter is named "bifurcation parameter", and in highly non-linear dynamic systems, its change can produce a large number of bifurcations between distinct dynamic modalities, with self-similarity and fractal structure. In many of these systems, we have a cascade of numberless bifurcations, culminating with the production of chaotic dynamics.

Chaotic Dynamics: Dynamics with specific features indicating complex behaviour, only produced in highly non-linear systems. These indicative features, formalized by the discipline of "Theory of Chaos", are high sensibility to initial conditions, non-periodic behaviour, and production of a large number of different trajectories in the state space, according to the change of some meaningful parameter of the dynamical system

C

(see bifurcation and diverse dynamics ahead). For some of the tools related to chaotic dynamics, see the related topic in the main text: Non-linear dynamics tools.

Chaotic Model Neurons: Model neurons that incorporate aspects of complex dynamics observed either in the isolated biological neuron or in assemblies of several biological neurons. Some of the models with complex dynamics, mentioned in the main text of this article, are the Aihara's model neuron, the Bifurcation Neuron proposed by Nabil Farhat, RPEs networks, Kaneko's CMLs, and Walter Freeman's K Sets.

Spatio-Temporal Collective Patterns: The observed dynamic configurations of the collective state variable in a multi neuron arrangement (network). The temporal aspect comes from the fact that in chaotic neural networks the model neurons' states evolve in time. The spatial aspect comes from the fact that the neurons that compose the network can be viewed as sites of a discrete (grid-like) spatial structure.

Stability: The study of repellers and attractors is done through stability analysis, which quantifies how infinitesimal perturbations in a given trajectory performed by the system are either attenuated or amplified with time.

Class Prediction in Test Sets with Shifted Distributions

Óscar Pérez
Universidad Autónoma de Madrid, Spain

Manuel Sánchez-Montañés
Universidad Autónoma de Madrid, Spain

INTRODUCTION

Machine learning has provided powerful algorithms that automatically generate predictive models from experience. One specific technique is *supervised learning*, where the machine is trained to predict a desired output for each input pattern $\mathbf{x}$. This chapter will focus on *classification*, that is, supervised learning when the output to predict is a class label. For instance predicting whether a patient in a hospital will develop cancer or not. In this example, the class label c is a variable having two possible values, "cancer" or "no cancer", and the input pattern $\mathbf{x}$ is a vector containing patient data (e.g. age, gender, diet, smoking habits, etc.). In order to construct a proper predictive model, supervised learning methods require a set of examples $\mathbf{x}_i$ together with their respective labels c_i. This dataset is called the "training set". The constructed model is then used to predict the labels of a set of new cases $\mathbf{x}_j$ called the "test set". In the cancer prediction example, this is the phase when the model is used to predict cancer in new patients.

One common assumption in supervised learning algorithms is that the statistical structure of the training and test datasets are the same (Hastie, Tibshirani & Friedman, 2001). That is, the test set is assumed to have the same attribute distribution $p(\mathbf{x})$ and same class distribution $p(c|\mathbf{x})$ as the training set. However, this is not usually the case in real applications due to different reasons. For instance, in many problems the training dataset is obtained in a specific manner that differs from the way the test dataset will be generated later. Moreover, the nature of the problem may evolve in time. These phenomena cause $p^{Tr}(\mathbf{x}, c) \neq p^{Test}(\mathbf{x}, c)$, which can degrade the performance of the model constructed in training.

Here we present a new algorithm that allows to re-estimate a model constructed in training using the unlabelled test patterns. We show the convergence properties of the algorithm and illustrate its performance with an artificial problem. Finally we demonstrate its strengths in a heart disease diagnosis problem where the training set is taken from a different hospital than the test set.

BACKGROUND

In practical problems, the statistical structure of training and test sets can be different, that is, $p^{Tr}(\mathbf{x}, c) \neq p^{Test}(\mathbf{x}, c)$. This effect can be caused by different reasons. For instance, due to biases in the sampling selection of the training set (Heckman, 1979; Salganicoff, 1997). Other possible cause is that training and test sets can be related to different contexts. For instance, a heart disease diagnosis model that is used in a hospital which is different from the hospital where the training dataset was collected. Then, if the hospitals are located in cities where people have different habits, average age, etc., this will cause a test set with a different statistical structure than the training set.

The special case $p^{Tr}(\mathbf{x}) \neq p^{Test}(\mathbf{x})$ and $p^{Tr}(c|\mathbf{x}) = p^{Test}(c|\mathbf{x})$ is known in the literature as "covariate shift" (Shimodaira, 2000). In the context of machine learning, the covariate shift can degrade the performance of standard machine learning algorithms. Different techniques have been proposed to deal with this problem, see for example (Heckman, 1979; Salganicoff, 1997; Shimodaira, 2000; Sugiyama, Krauledat & Müller, 2007). Transductive learning has also been suggested as another way to improve performance when the statistical structure of the test set is shifted with respect to the training set (Vapnik, 1998; Chen, Wang & Dong, 2003; Wu, Bennett, Cristianini & Shawe-Taylor, 1999).

The statistics of the patterns $\mathbf{x}$ can also change in time, for example in a company that has a continuous

Figure 1. Changes across time of the statistics of clients in a car insurance company. The histograms of two different variables (a, b) related to the clients' use of their insurance are shown. Dash: data collected four months later than data shown in solid.

C
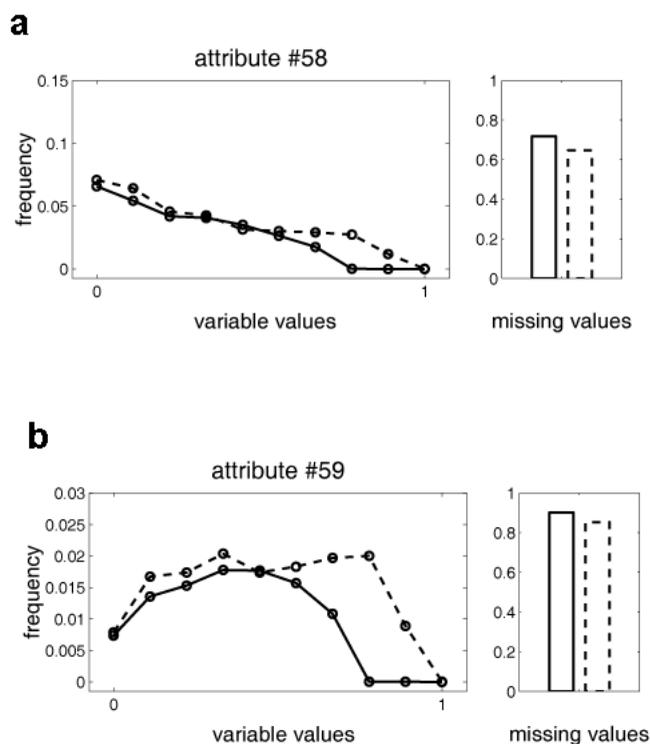
flow of new and leaving clients (figure 1). If we are interested in constructing a model for prediction, the statistics of the clients when the model is exploited will differ from the statistics in training. Finally, often the concept to be learned is not static but evolves in time (for example, predicting which emails are spam or not), causing $p^{Tr}(\mathbf{x}, c) \neq p^{Test}(\mathbf{x}, c)$. This problem is known as "concept drift" and different algorithms have been proposed to cope with it (Black & Hickey, 1999; Wang, Fan, Yu, & Han, 2003; Widmer & Kubat, 1996).

A NEW ALGORITHM FOR CONSTRUCTING CLASSIFIERS WHEN TRAINING AND TEST SETS HAVE DIFFERENT DISTRIBUTIONS

Here we present a new learning strategy for problems where the statistical distributions of the training and test sets are different. This technique can be used in problems where concept drift, sampling biases, or any other phenomena exist that cause the statistical structure of the training and test sets to be different. On the other hand, our strategy constructs an explicit estimation of the statistical structure of the problem in the test data set. This allows us to construct a classifier that is optimized with respect to the new test statistics, and provides the user with relevant information about which aspects of the problem have changed.

Algorithm

1. Construct a statistical model { $\tilde{P}(x|c)$, $\tilde{P}(c)$ } for the training set using a standard procedure (for example, using the standard EM algorithm).
2. Re-estimate this statistical model using the non-labelled patterns **x** of the test set. For this purpose, we have developed a semi-supervised extension of EM.

3. Use this re-estimated statistical model $\{\tilde{P}'(x|c),$ $\tilde{P}'(c)\}$ to construct a classifier optimized to the test set.

Model Re-Estimation: A Semi-Supervised Extension of EM

The standard EM algorithm (Dempster, Laird & Rubin, 1977) is an iterative procedure designed to find the optimal parameters of a statistical model in a maximum likelihood sense. Here we present an extension of the EM algorithm that re-estimates the statistical model learned in training using the unlabelled test set, under the assumption that it should resemble the statistical model learned at training. That is, the algorithm finds the minimum amount of change that one has to assume for the model constructed in training (where we know the pattern classes) to explain the global distribution of attributes **x** in the test set.

Let us call Θ the set of different parameters in the statistical model of our problem. For example, if we model $\tilde{P}(\mathbf{x}|c=1)$ and $\tilde{P}(\mathbf{x}|c=2)$ by a mixture of two and three Gaussians respectively, then Θ would be composed by the averages, covariance matrices and likelihoods of the 2+3 different Gaussians in the model. An estimation Θ_{Tr} should be first made in the training set using a standard technique such as applying EM for each individual class. Then, we should recalculate it as Θ_{Te} using the unlabelled test set by optimizing the likelihood $\tilde{P}(\Theta_{Te}|\mathcal{D}_{Te},\Theta_{Tr})$ where $\mathcal{D}_{Te}$ is the unlabelled test set (test patterns without the class information). The maximization of this quantity respect to Θ_{Te} is equivalent to maximizing the quantity

$$L' \equiv \ln \tilde{P}(\mathcal{D}_{Te}|\Theta_{Te}) + \ln \tilde{P}(\Theta_{Te}|\Theta_{Tr})$$

which we will call the "extended log-likelihood". The term $\tilde{P}(\Theta_{Te}|\Theta_{Tr})$ implements the bias in the re-estimation of the parameters, which in our case is a preference for small changes. Using this extended log-likelihood it is possible to derive a new version of EM that maximizes L'.

To achieve it we consider, as in standard EM applications, the existence of additional but *latent* (or 'hidden') variables **h** in the problem (Dempster, Laird & Rubin, 1977). For example, in case we model the statistics as a mixture of Gaussians, the latent variable indicates which of the Gaussians actually generated the pattern. The parameters of our statistical model are then of two types: those α affecting the probability distributions of the hidden variables **h**, and those β affecting the rest of parameters of the model. Therefore, $\Theta = \{\alpha, \beta\}$. For instance, if the statistical model is a mixture of Gaussians, α contains the prior probabilities of the different Gaussians, and β is composed by the averages and covariance matrices. Finally, we assume that the penalization term can be written as:

$$\ln \tilde{P}(\Theta_{Te}|\Theta_{Tr}) = \ln \tilde{P}(\alpha_{Te}|\alpha_{Tr}) + \ln \tilde{P}(\beta_{Te}|\beta_{Tr})$$

We are now in a position to develop the semi-supervised extension of EM. Following the same scheme of reasoning as in exercise 44 of chapter 3 in (Duda, Hart & Stork, 2001) we arrive at the following algorithm:

1. Initialize $\alpha'_{Te} \leftarrow \alpha_{Tr}$ and $\beta'_{Te} \leftarrow \beta_{Tr}$
2. (**E** step): Compute for all **h** and $\mathbf{x}_i$:

$$\tilde{P}(\mathbf{h}|\mathbf{x}_i,\alpha'_{Te},\beta'_{Te}) = \frac{\tilde{P}(\mathbf{x}_i|\mathbf{h},\beta'_{Te}) \cdot \tilde{P}(\mathbf{h}|\alpha'_{Te})}{\sum_{\mathbf{h}'} \tilde{P}(\mathbf{x}_i|\mathbf{h}',\beta'_{Te}) \cdot \tilde{P}(\mathbf{h}'|\alpha'_{Te})}$$

3. (**M** step):
   Calculate the $\alpha*_{Te}$ that maximizes the following quantity (fixing α'_{Te} and β'_{Te}):

$$\ln \tilde{P}(\alpha*_{Te}|\alpha_{Tr}) + \sum_{\mathbf{h},\mathbf{x}_i} \tilde{P}(\mathbf{h}|\mathbf{x}_i,\alpha'_{Te},\beta'_{Te}) \cdot \ln \tilde{P}(\mathbf{h}|\alpha*_{Te})$$

   Calculate the $\beta*_{Te}$ that maximizes the following quantity (fixing α'_{Te} and β'_{Te}):

$$\ln \tilde{P}(\beta*_{Te}|\beta_{Tr}) + \sum_{\mathbf{h},\mathbf{x}_i} \tilde{P}(\mathbf{h}|\mathbf{x}_i,\alpha'_{Te},\beta'_{Te}) \cdot \ln \tilde{P}(\mathbf{x}_i|\mathbf{h},\beta*_{Te})$$

4. Update the parameters: $\alpha'_{Te} \leftarrow \alpha*_{Te}$ and $\beta'_{Te} \leftarrow \beta*_{Te}$
5. Go to step 2 until convergence of L'

In this derivation it is also guaranteed that L' does not decrease at each step, so our algorithm will find at least a local optimum of L'. On the other hand, the penalization term will ensure the stability of the algorithm (small changes in test data lead to small changes in the re-estimation), and will allow to associate the correct class to the different clusters. Our algorithm contains as a special case the standard EM algorithm

C

when the distribution $\tilde{P}(\Theta_{Te}|\Theta_{Tr})$ is not considered or is assumed to be homogeneous.

In the examples we will show in this chapter we use a simple case of this algorithm where the statistical model consists in one Gaussian per class, and only the averages of the Gaussians are re-estimated. The penalization term $\ln \tilde{P}(\Theta_{Te}|\Theta_{Tr})$ used in these examples is proportional to the Mahalanobis distance (Duda, Hart & Stork, 2001) between the averages estimated in training and those re-estimated in test (we will refer to the proportional factor as γ). Then we arrive at the following simplified algorithm:

1. Initialize $\mu'_{1,Te} \leftarrow \mu_{1,Tr}$ and $\mu'_{2,Te} \leftarrow \mu_{2,Tr}$
2. (E step): Compute for c=1,2 and all $\mathbf{x}_i$ in the test dataset:

$$\tilde{P}(c|\mathbf{x}_i) = \frac{\tilde{P}(c) \cdot \tilde{p}(\mathbf{x}_i|\mu'_{c,Te}, \mathbf{M}_c)}{\tilde{P}(1) \cdot \tilde{p}(\mathbf{x}_i|\mu'_{1,Te}, \mathbf{M}_1) + \tilde{P}(2) \cdot \tilde{p}(\mathbf{x}_i|\mu'_{2,Te}, \mathbf{M}_2)}$$

3. (M step):

$$\mu'_{c,Te} \leftarrow \frac{\gamma \cdot (\mu_{c,Tr} + \mathbf{d}) + \frac{1}{N_{Te}}\sum_{\mathbf{x}_i} \tilde{P}(c|\mathbf{x}_i) \cdot \mathbf{x}_i}{\gamma + \frac{1}{N_{Te}}\sum_{\mathbf{x}_i} \tilde{P}(c|\mathbf{x}_i)}$$

and go to step 2 until convergence of L'.

where $c \in \{1, 2\}$ is the class of the pattern; $\tilde{P}(c)$ is the prior probability of class c estimated in the training set; $\mu_{c,Tr}$ and $\mu_{c,Te}$ are the averages of the patterns of class c in training and test sets respectively; $\mathbf{M}_c$ is the covariance matrix of class c estimated in the training set; $\tilde{p}(\mathbf{x}|\mu, \mathbf{M})$ is the probability density function of $\mathbf{x}$ given that it has been generated by a Gaussian process of average μ and covariance matrix $\mathbf{M}$; $\mathbf{d}$ is the substraction of the global average of the attributes in test to the global average of the attributes in training; and N_{Te} is the number of patterns in test.

Application to a Synthetic Problem

First we will illustrate our algorithm using a simple synthetic problem with two classes (figure 2). In training, class "grey" was generated from a Gaussian distribution with average [1.0 ; 1.0] and covariance matrix [0.5, 0.0

; 0.0, 0.5]; class "black" was generated from a Gaussian distribution with average [1.0 ; 1.5] and covariance matrix [0.5, 0.4 ; 0.4, 0.5]. The statistical structure of the test set is different than in training due to a shift of class "black" (figure 2a and 2b), now with average [2.0 ; 1.0]. The minimum Bayes' errors of training and test sets are 27.7% and 18.8% respectively.

First we construct a statistical model of the training set consisting in one Gaussian for each class (figure 2c). This statistical model is then used to construct a classifier based on Linear Discriminant Analysis (LDA). The error of this classifier in training is 34.1%. When applied to test set, the error of this classifier increases to 66.0%. Our algorithm was then used to re-estimate the statistical model constructed in training using the unlabelled test set patterns (figure 2d). We can observe that our algorithm finds the appropriate statistical model for the test set (figure 2e). In fact, when we recalculated the LDA classifier using the re-estimated statistical model, the error in test decreased to 20.5%.

Application to Heart Disease Diagnosis when the Training and Test Hospitals are Different

We tested our algorithm using the Heart disease database from the UCI machine learning repository (Asuncion & Newman, 2007). The goal is to predict whether a patient has a heart disease or not, given some personal data (age, sex, smoking habits, etc.) and the results of several medical examinations such as blood pressure and electro cardiograms. The Heart disease database is in fact the union of four datasets, each one obtained at a different hospital. One would expect that the statistical structure of the problem is different in different hospitals since the patients live in different cities (and thus the environmental factors can be different), the measurement devices are not exactly the same, etc. Thus we expect that our algorithm will improve the classification performance of a model constructed in a different hospital.

We have checked this hypothesis using the data from the Cleveland Clinic Foundation as training, and the data from the Hungarian Institute of Cardiology as test. In both cases we removed the attributes in columns 12 and 13 since they are frequently missing in the Hungarian dataset, and performed a normalization in each set so the attributes have zero mean and unit variance. The result is 297 examples from the Cleveland Clinic (149

Figure 2. Re-estimation of the statistical model of the problem in a synthetic example. The two different classes are shown as two different clouds (grey and black). a, b: the statistical structure of the problem in training is different than in test. c: statistical model constructed from the training set (dashed: model for class "grey"; solid: model for class "black"). d: recalculation of the statistical model using the unlabelled test set (note that we have drawn all patterns in grey since the re-estimation algorithm does not have access to the classes of the test patterns). e: Re-estimated statistical model superposed with the labelled test set (dashed: model for class "grey"; solid: model for class "black").
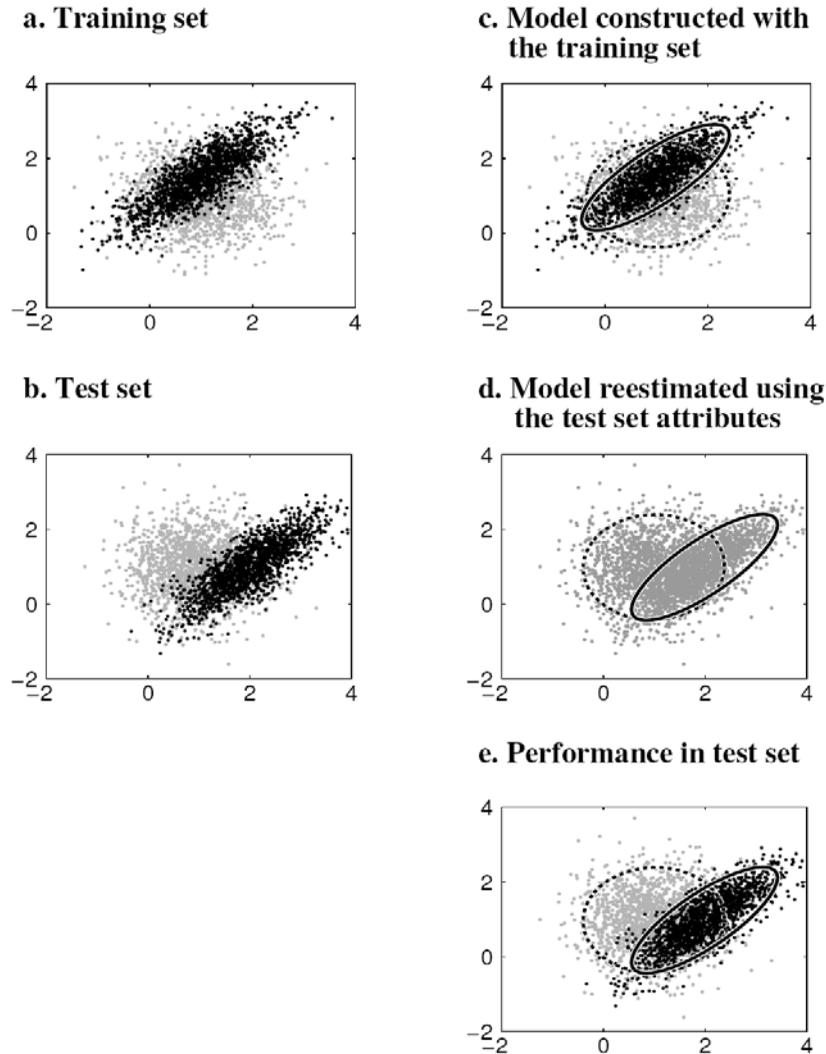
of class 1, 148 of class 2), and 294 from the Hungarian Institute of Cardiology (188 of class 1, 106 of class 2). First, we use PCA to reduce the dimensionality of the problem in the training set. Therefore the statistical model of our problem now consists of the average and covariance matrix of each class in the principal axes. Then, a simple classifier that assigns to pattern **x** the class with nearest center is considered. The number of principal components is automatically selected on the basis of the performance of the classifier in a random subset of the training dataset which was reserved for this task. Once the statistical model of the training set is constructed, we evaluated the performance of the classifier in the test set obtaining an error rate of 18.7%.

In a second step we used our algorithm for re-estimating the statistical model using $\gamma=0.01$, and then

C

classified the test patterns using the same procedure as before, that is, assigning to each test pattern the class with nearest center. The error was reduced in this case to 15.3 %. If we repeat the same strategy using a more elaborated classifier such as one based on Linear Discriminant Analysis, we observe a reduction in the error from 17.0% to 13.9% after using our algorithm.

FUTURE TRENDS

In the examples we have presented here we have used a simplified version of our algorithm that assumes a statistical model consisting of one Gaussian for each class, and that only the averages need to be re-estimated with test. However, our semi-supervised extension of EM is a general algorithm that is applicable with arbitrary parametric statistical models (e.g. mixtures of an arbitrary number of non Gaussian models), and allows to re-estimate any parameter of the model. Future work will include the study of the performance and robustness of different types of statistical models in practical problems. On the other hand, since our approach is based on an extension of the extensively studied standard EM algorithm, we expect that analytical results such as generalization error bounds can be derived, making the algorithm attractive also from a theoretical perspective.

CONCLUSION

We have presented a new learning strategy for classification problems where the statistical structure of training and test sets are different. In this kind of problems, the performance of traditional machine learning algorithms can be severely degraded. Different techniques have been proposed to cope with different aspects of the problem, such as strategies for addressing the sample selection bias (Heckman, 1979; Salganicoff, 1997; Shimodaira, 2000; Zadrozny, 2004; Sugiyama, Krauledat & Müller, 2007), strategies for addressing the concept drift (Black & Hickey, 1999; Wang, Fan, Yu & Han, 2003; Widmer & Kubat, 1996) and transductive learning (Vapnik, 1998; Wu, Bennett, Cristianini & Shawe-Taylor, 1999).

The learning strategy we have presented allows to address all these different aspects, so that it can be used in problems where concept drift, sampling biases, or any

other phenomena exist that cause the statistical structure of training and test sets to be different. Moreover, our algorithm constructs an explicit statistical model of the new structure in the test dataset, which is of great value for understanding the dynamics of the problem and exploiting this knowledge in practical applications. To achieve this we have extended the EM algorithm obtaining a semi-supervised version that allows to re-estimate the statistical model constructed in training using the attribute distribution of the unlabelled test set patterns.

REFERENCES

Asuncion, A., & Newman, D.J. (2007). *UCI Machine Learning Repository*. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science

Black, M.M., & Hickey, R.J. (1999). Maintaining the Performance of a Learned Classifier under Concept Drift. *Intelligent Data Analysis*. (3) 453-474.

Chen, Y., Wang, G., & Dong, S. (2003). Learning with Progressive Transductive Support Vector Machine. *Pattern Recognition Letters*. (24) 1845-1855.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Royal Statist. Soc. Ser. B*. (39) 1-38.

Duda, R.O., Hart, P.E., & Stork, D.G. (2001). *Pattern Classification* (2nd ed.). John Wiley & Sons.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Heckman, J.J. (1979). Sample Selection Bias as a Specification Error. *Econometrica*. (47) 153-161.

Salganicoff, M. (1997). Tolerating Concept and Sampling Shift in Lazy Learning Using Prediction Error Context Switching. *Artificial Intelligence Review*. (11) 133-155.

Shimodaira, H. (2000). Improving Predictive Inference under Covariate Shift by Weighting the Log-Likelihood Function. *Journal of Statistical Planning and Inference*. (90) 227-244.

Sugiyama, M., Krauledat, M., Müller, K.-R. (2007). Covariate Shift Adaptation by Importance Weighted Cross Validation. *Journal of Machine Learning Research*. (8) 985-1005.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

Wang, H., Fan, W., Yu, P.S., & Han, J. (2003). Mining Concept-Drifting Data Streams Using Ensemble Classifiers. In Proc. 9th Int. Conf. on Knowledge Discovery and Data Mining KDD-2003.

Widmer, G., & Kubat, M. (1996). Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*. (23) 69-101.

Wu, D., Bennett, K., Cristianini, N., & Shawe-Taylor, J. (1999). Large Margin Trees for Induction and Transduction. In Proc. 16th International Conf. on Machine Learning.

Zadrozny, B. (2004). Learning and Evaluating Classifiers under Sample Selection Bias. In Proc. 21st International Conference on Machine Learning.

KEY TERMS

Attribute: Each of the components that constitute an input pattern.

Classifier: Function that associates a class c to each input pattern $\mathbf{x}$ of interest. A classifier can be directly constructed from a set of pattern examples with their respective classes, or indirectly from a statistical model $\widetilde{P}(\mathbf{x}, c)$.

EM (Expectation-Maximization Algorithm): standard iterative algorithm for estimating the parameters Θ of a parametric statistical model. EM finds the specific parameter values that maximize the likelihood of the observed data D given the statistical model, $\widetilde{P}(D \mid \Theta)$. The algorithm alternates between the Expectation step and the Maximization step, finishing when $\widetilde{P}(D \mid \Theta)$ meets some convergence criterium.

Missing Value: Special value of an attribute that denotes that it is not known or can not be measured.

Semi-Supervised Learning: Machine learning technique that uses both labelled and unlabelled data for constructing the model.

Statistical Model: Mathematical function that models the statistical structure of the problem. For classification problems, the statistical model is $\widetilde{P}(\mathbf{x}, c)$ or equivalently $\{ \widetilde{P}(\mathbf{x} \mid c), \widetilde{P}(c) \}$ since $\widetilde{P}(\mathbf{x}, c) = \widetilde{P}(\mathbf{x} \mid c) \cdot \widetilde{P}(c)$.

Supervised Learning: Type of learning where the objective is to learn a function that associates a desired output ('label') to each input pattern. Supervised learning techniques require a training dataset of examples with their respective desired outputs. Supervised learning is traditionally divided into regression (the desired output is a continuous variable) and classification (the desired output is a class label).

Training/Test Sets: In the context of this chapter, the training set is composed by all labelled examples that are provided for constructing a classifier. The test set is composed by the new unlabelled patterns whose classes should be predicted by the classifier.

Cluster Analysis of Gene Expression Data

Alan Wee-Chung Liew
Griffith University, Australia

Ngai-Fong Law
The Hong Kong Polytechnic University, Hong Kong

Hong Yan
City University of Hong Kong, Hong Kong
University of Sydney, Australia

INTRODUCTION

Important insights into gene function can be gained by gene expression analysis. For example, some genes are turned on (expressed) or turned off (repressed) when there is a change in external conditions or stimuli. The expression of one gene is often regulated by the expression of other genes. A detail analysis of gene expression information will provide an understanding about the inter-networking of different genes and their functional roles.

DNA microarray technology allows massively parallel, high throughput genome-wide profiling of gene expression in a single hybridization experiment [Lockhart & Winzeler, 2000]. It has been widely used in numerous studies over a broad range of biological disciplines, such as cancer classification (Armstrong et al., 2002), identification of genes relevant to a certain diagnosis or therapy (Muro et al., 2003), investigation of the mechanism of drug action and cancer prognosis (Kim et al., 2000; Duggan et al., 1999). Due to the large number of genes involved in microarray experiment study and the complexity of biological networks, clustering is an important exploratory technique for gene expression data analysis. In this article, we present a succinct review of some of our work in cluster analysis of gene expression data.

BACKGROUND

Cluster analysis is a fundamental technique in exploratory data analysis (Jain & Dubes, 1988). It aims at finding groups in a given data set such that objects in the same group are similar to each other while objects in different groups are dissimilar. It aids in the discovery of gene function because genes with similar gene expression profiles can be an indicator that they participate in related cellular processes. Clustering of genes may suggest possible roles for genes with unknown functions based on the known functions of some other genes in the same cluster. Clustering of gene expression data has been applied to, for example, the study of temporal expression of yeast genes in sporulation (Chu et al., 1998), the identification of gene regulatory networks (Chen, Filkov, & Skiena, 1999), and the study of cancer (Tamayo et al., 1999).

Many clustering algorithms have been applied to the analysis of gene expression data (Sharan, Elkon, & Shamir, 2002). They can be broadly classified as either hierarchical or partition-based depending on how they group the data. Hierarchical clustering is further subdivided into agglomerative methods and divisive methods. The former proceed by successive merging of the N objects into larger groups, whereas the latter divide a larger group successively into finer groupings. Agglomerative techniques are more common in hierarchical clustering.

Hierarchical clustering is among the first clustering technique being applied to gene expression data (Eisen et al., 1998). In hierarchical clustering, each of the gene expression profile is considered as a cluster initially. Then, pairs of clusters with the smallest distance between them, are merged together to form a single cluster. This process is repeated until there is only one cluster left. The hierarchical clustering algorithm arranges the gene expression data into a hierarchical tree structure known as a dendrogram, which allows easy visualization and interpretation of results. However, the hierarchical tree cannot indicate the optimal number of clusters in the data. The user has to interpret the tree topologies and identify branch

points that segregate clusters of biological relevance. In addition, once a data is assigned to a node in the tree, it cannot be reassigned to a different node even though it is later found to be closer to that node.

In partition-based clustering algorithms, such as K-means clustering (Jain & Dubes, 1988), the number of clusters is arbitrarily fixed by the users at start. Setting the correct number of clusters can be a difficult problem and many heuristics are used. The basic idea of K-means clustering is to partition the data into a predefined number of clusters such that the variability of the data within each cluster is minimized. Clustering is achieved by first generating K random cluster centroids, then alternatingly updating the cluster assignment of each data vector and the cluster centroids. The Euclidean distance is usually employed in K-means clustering to measure the closeness of a data vector to the cluster centroids. However, such distance metric inevitably imposes an ellipsoidal structure on the resulting clusters. Hence, data that do not conform to this structure are poorly clustered by the K-means algorithm.

Other approach to clustering includes model-based approach. In contrast to model-free partition-based algorithms, model-based clustering uses certain distribution models for clusters and attempts to optimize the fit between the data and the model. Each cluster is represented by a parametric distribution, like a Gaussian, and the entire data set is modeled by a mixture of these distributions. The most widely used clustering method of this kind is the one based on a mixture of Gaussians (McLachlan & Basford, 1988; Yeung et al., 2001).

CLUSTERING OF GENE EXPRESSION DATA

Binary Hierarchical Clustering (BHC): In Szeto et al. (2003), we proposed the BHC algorithm for clustering gene expression data based on the hierarchical binary subdivision framework of Clausi (2002). Consider the dataset with three distinct classes as shown in Fig.1. The algorithm starts by assuming that the data consists of one class. The first application of binary subdivision generates two clusters A and BC. As the projection of class A and class BC have a large enough Fisher criterion on the A-BC discriminant line, the algorithm splits the original dataset into two clusters. Then, the binary subdivision is applied onto each of the two clusters. The

Figure 1. The binary subdivision framework. (a) Original data treated as one class, (b) Partition into two clusters, A and BC. (c) Cluster A cannot be split further, but cluster BC is split into two clusters, B and C. (d) Both cluster B and C cannot be split any more, and we have three clusters A, B, and C. (Figure adopted from Clausi (2002))
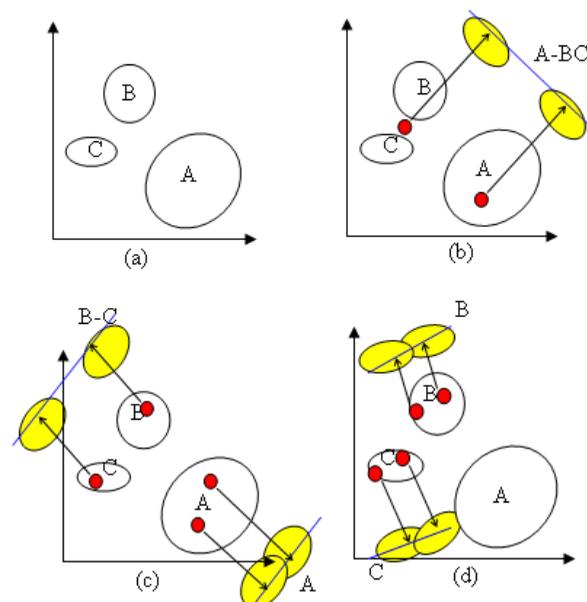
BC cluster is separated into B and C clusters because its Fisher criterion is large. However, the Fisher criterion of cluster A is too small to allow further division, so it remains as a single cluster. Such subdivision process is repeated until all clusters have Fisher criterion too low for further splitting and the process stops.

The BHC algorithm proceeds in two steps: (1) binary partition using a mixed FCM-HC algorithm to partition the data into two classes, (2) Fisher discriminant analysis on the two classes, where if the Fisher criterion exceeds a set threshold, we accept the partition; otherwise do not subdivide the data any further.

A novel aspect of the BHC algorithm is the use of the FCM-HC algorithm to partition the data. The idea is illustrated in Fig.2a-2c. In Fig.2a, there are three clusters. The desirable two-class partition would be for the two closer clusters to be classified as one partition and the remaining cluster as the second partition. However, a K-means partitioning of the data would favor the split of the middle cluster into two classes. With this wrong partition, subsequent Fisher discriminant analysis would conclude that the splitting is not valid and the data should be considered as just one class. To overcome this problem, we first over-cluster the data into several clusters by the Fuzzy C-means (FCM) algorithm. Then, the clusters are merged together by the average linkage hierarchical clustering (HC) algorithm until only two classes are left. In Fig.2a, the data is over-clustered into six clusters by the FCM algorithm. A hierarchical tree constructed from the six clusters using the average linkage clustering algorithm is constructed as in Fig.2b. Using the cutoff as shown in

Fig.2b, a final partitioning of the data into two classes is shown in Fig.2c. We can see that A, B are merged into one class, and C, E, F, D are merged into the second class. FCM-HC also allows non-ellipsoidal clusters to be partitioned correctly. Fig.2d shows a dataset containing two non-ellipsoidal clusters that is over-clustered into six clusters. After merging, we obtain the correct two-class partitions as shown in Fig.2e.

The BHC algorithm makes no assumption about the class distributions and the number of clusters is determined automatically. The only parameter required is the Fisher threshold. The binary hierarchical framework naturally leads to a tree structure representation where similar clusters are placed adjacent to each other in the tree. Near the root of the tree, only gross structures in the data are shown, whereas near the end branches, fine details in the data can be visualized. Figure 3 shows the BHC clustering results on Spellman's yeast dataset (http://cellcycle-www.stanford.edu). The dataset contains expression profiles for 6178 genes under different experimental conditions, i.e., cdc15, and cdc28, alpha factor and elutriation experiments.

Self-Splitting and Merging Competitive Learning (SSMCL) Clustering: In Wu, Liew, & Yan (2004), we proposed a new clustering framework based on the one-prototype-take-one-cluster (OPTOC) learning paradigm (Zhang & Liu, 2002) for clustering gene expression data. The new algorithm is able to identify natural clusters in the data as well as provide a reliable estimate of the number of distinct clusters in the data. In conventional clustering, if the number of

Figure 2. Partition of the data into two classes, where the original data is clustered into six clusters (a) with the resulting hierarchical tree (b), and finally the two-class partition after merging (c). For two non-ellipsoidal clusters, FCM-HC over-clusters them into 6 classes (d), and the resulting two-class partition after merging (e).
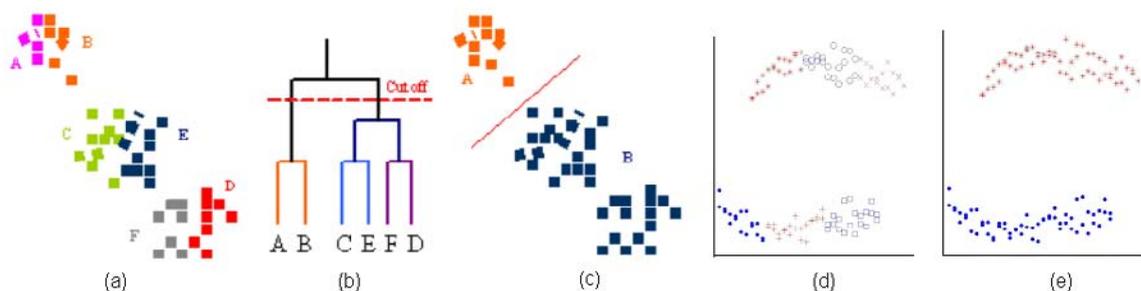
Figure 3. BHC clustering of (a) alpha factor experiment dataset, (b) cdc15 experiment dataset, (c) elutriation experiment dataset, and (d) cdc28 experiment dataset.
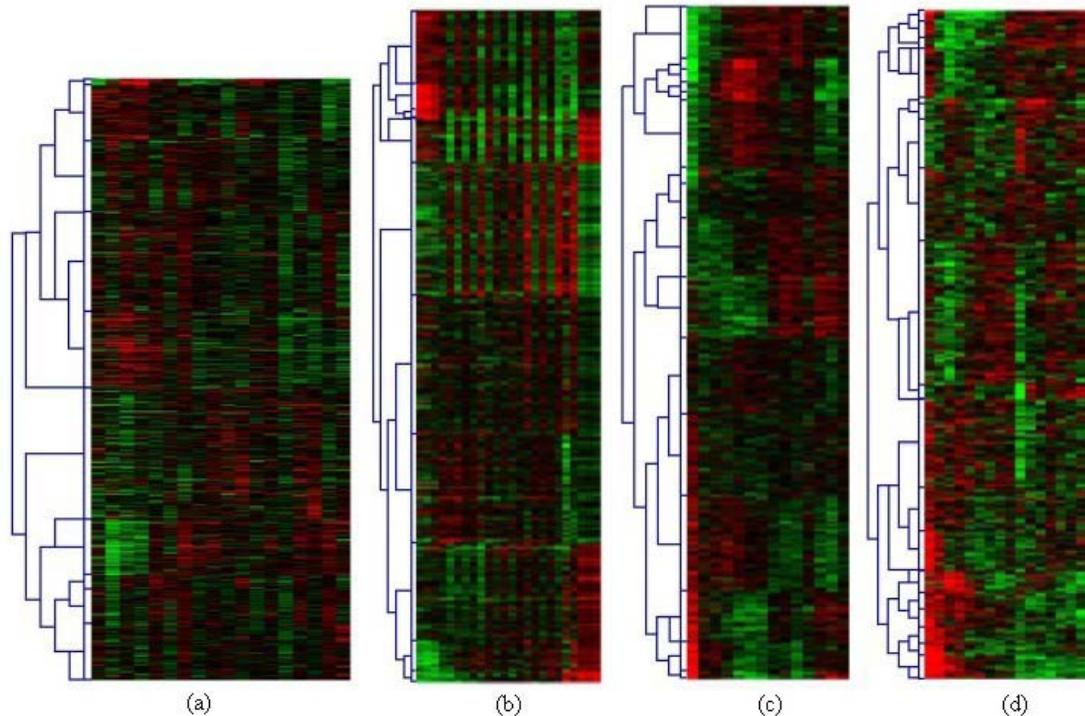
prototypes is less than that of the natural clusters in the data, then at least one prototype will win patterns from more than two clusters. In contrast, the OPTOC idea allows one prototype to characterize only one natural cluster in the data, regardless of the actual number of clusters in the data. This is achieved by constructing a dynamic neighborhood such that patterns inside the neighborhood contribute more to its learning than those outside. Eventually, the prototype settles at the center of a natural cluster, while ignoring competitions from other clusters as shown in Fig. 4b.

The SSMCL algorithm starts with a single cluster. If the actual number of clusters in the data is more than one, additional prototypes are generated to search for the remaining clusters. Let C_i denotes the center of all the patterns that P_i wins according to the minimum distance rule. The distortion $|P_i-C_i|$ measures the discrepancy between the prototype P_i found by OPTOC learning and the actual cluster structure in the data. For example, in Fig. 4b, C_1 would be located at the center of the three clusters S1, S2 and S3 (since there is only one prototype, it wins all input patterns), while P_1 eventually settled at the center of S3. After the prototypes have all settled down, a large $|P_i-C_i|$ indicates the presence of other natural clusters in the data. A new prototype would be generated from the prototype with the largest distortion when this distortion exceeds a certain threshold ε.

Ideally, with a suitable threshold, the algorithm will find all natural clusters in the data. Unfortunately, the complex structure exhibited by gene expression data makes setting an appropriate threshold difficult. Instead,

Figure 4. Two learning methods: conventional versus OPTOC. (a) One prototype takes the arithmetic center of three clusters (conventional learning). (b) One prototype takes one cluster (OPTOC learning) and ignores the other two clusters.

(a)   *(b)*
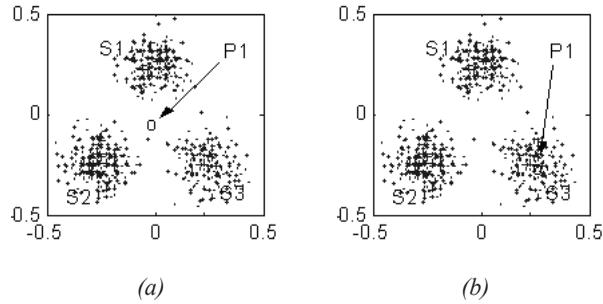
Figure 5. (a) The 22 distinct clusters found by SSMCL for the yeast cell cycle data. (b) Five patterns that correspond to the five cell cycle phases. The genes presented here are only those that belong to these clusters and are biologically characterized to a specific cell cycle phase.
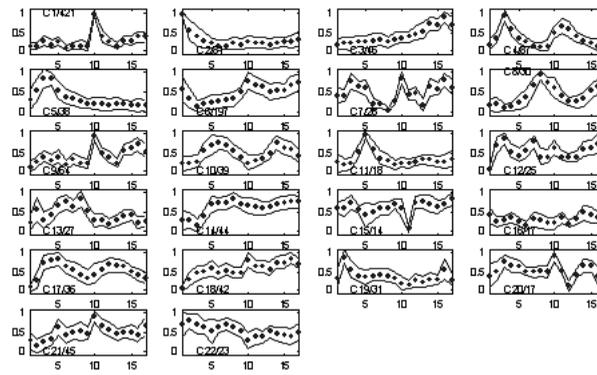
(a)
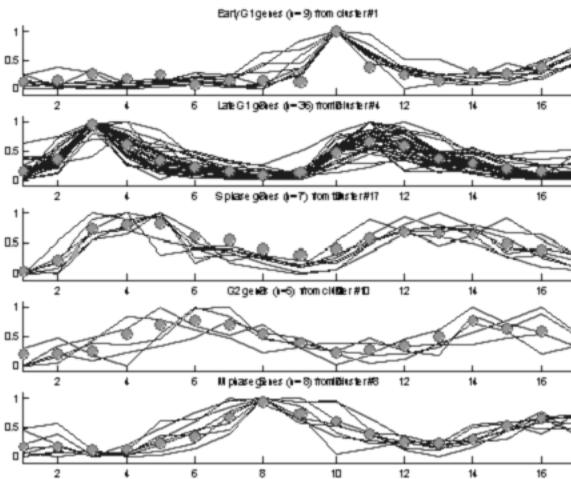
(b)

we proposed an over-clustering and merging strategy. The over-clustering step minimizes the chance of missing any natural clusters in the data, while the merging step ensures that the final clusters are all visually distinct from each other. In over-clustering, a natural cluster might be split into more than one cluster. However, no one cluster may contain data from several natural clusters, since the OPTOC paradigm discourages a cluster from winning data from more than one natural cluster. The clusters that are visually similar would be merged during the merging step. Together with the OPTOC framework, the over-clustering and merging framework allows a systematic estimation of the correct number of natural clusters in the data.

Fig.5a shows the SSMCL clustering result for the yeast cell cycle expression data (http://genomics. stanford.edu). We observe that the 22 clusters found have no apparent visual similarity. We checked the 22 clusters with the study of Cho et al. (1998), where 416 genes have been interpreted biologically. Those gene expression profiles include five fundamental patterns that correspond to five cell cycles phases: early G1, late G1, S, G2, and M phase. Fig.5b shows the five clusters that contain most of the genes belonging to these five different patterns. It is obvious that these five clusters correspond to the five cell cycle phases.

FUTURE TRENDS

Microarray data is usually represented as a matrix with rows and columns correspond to genes and conditions respectively. Conventional clustering algorithms can be applied to either rows or columns but not simultaneously. However, an interesting cellular process is often active only in a subset of conditions, or a single gene may participate in multiple pathways that may not be co-active under all conditions. Biclustering methods allow the clustering of rows and columns simultaneously. Existing biclustering algorithms often iteratively search for the best possible sub-grouping of the data by permuting rows and columns of the data matrix such that an appropriate merit function is improved (Madeira & Oliveira, 2004). When different bicluster patterns co-exist in the data, no single merit function can adequately cater for all possible patterns. Although recent approach such as geometric biclustering (Gan, Liew & Yan, 2008) has shown great potential, much work is still needed here. It is also important to de-

velop visualizing tools for the high dimensional gene expression clusters. The parallel coordinate plot is a well-known visualization technique for high dimension data and we are currently investigating it for bicluster visualization (Cheng et al., 2007).

CONCLUSION

Cluster analysis is an important exploratory tool for gene expression data analysis. This article describes two recently proposed clustering algorithms for gene expression data. In the BHC algorithm, the data are successively partitioned into two classes using the Fisher criterion. The binary partitioning leads to a tree structure representation which facilitates easy visualization. In the SSMCL algorithm, the OPTOC learning framework allows the detection of natural clusters. The subsequent over-clustering and merging step then allows a systematic estimation of the correct number of clusters in the data. Finally, we discuss some possible avenues of future research in this area. The problem of biclustering of gene expression data is a particularly interesting topic that warrants further investigation.

ACKNOWLEDGMENT

REFERENCES

Armstrong, S.A, Staunton, J.E., Silverman, L.B, Pieters, R., den Boer, M.L., Minder, M.D., Sallan, S.E., Lander, E.S., Golub, T.R. & Korsmeyer, S.J. (2002). MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*. 30: 41-47.

Chen, T., Filkov, V., & Skiena, S.S. (1999). Identifying gene regulatory networks from experimental data. *Proceedings of the Third Annual International Conference on Computational Molecular Biology RECOMB99*, Lyon, France. 94-103.

Cheng, K.O., Law, N.F., Siu, W.C., & Lau, T.H. (2007). BiVisu: Software Tool for Bicluster Detection and

Visualization**.** *Bioinformatics*. doi: 10.1093/bioinformatics/btm338.

Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D., Lockhart D.J., & Davis, R.W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*. 2(1): 65-73.

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O., & Herskowitz, I. (1998). The transcriptional program of sporulation in budding yeast. *Science*. 282: 699-705.

Clausi, D.A. (2002). K-means Iterative Fisher (KIF) unsupervised clustering algorithm applied to image texture segmentation. *Pattern Recognition*. 35(9): 1959-1972.

Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P. & Trent, J.M. (1999). Expression profiling using cDNA microarrays. *Nature Genetics*. 21: 10-14.

Eisen, M.B., Spellman, P.T., Brown, P.O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, 95(25): 14863-14868.

Gan, X., Liew, A.W.C., & Yan, H. (2008). Discovering biclusters in gene expression data based on high-dimensional linear geometries. BMC Bioinformatics, 9:209, doi: 10.1186/1471-2105-9-209.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., & Lander, E.S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*. 286(5439): 531-537.

Jain, A.K. & Dubes, R.C. (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, N.J.

Kim, S., Dougherty, E.R., Chen, Y., Sivakumar, K., Meltzer, P., Trent, J.M. and Bittner, M. (2000) Multivariate measurement of gene expression relationships. *Genomics*. 67: 201-209.

Lockhart, D.J., & Winzeler, E.A. (2000). Genomics, gene expression and DNA arrays. *Nature*. 405: 827–846.

Madeira, S.C. & Oliveira, A.L. (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 1 (1): 24-45.

McLachlan, G.J., Basford, K.E. (1988). *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker.

Muro, S., Takemasa, I., Oba, S., Matoba, R., Ueno, N., Maruyama, C., Yamashita, R., Sekimoto, M., Yamamoto, H., Nakamori, S., Monden, M., Ishii, S. & Kato, K. (2003). Identification of expressed genes linked to malignancy of human colorectal carcinoma by parametric clustering of quantitative expression data. *Genome Biology*. 4: R21

Sharan, R., Elkon, R., & Shamir, R. (2002). Cluster analysis and its applications to gene expression data. *Ernst Schering Workshop on Bioinformatics and Genome Analysis*. Berlin: Springer-Verlag, 83-108.

Szeto, L.K., Liew, A.W.C., Yan, H., & Tang, S.S. (2003). Gene expression data clustering and visualization based on a binary hierarchical clustering framework. *Special issue on Biomedical Visualization for Bioinformatics, Journal of Visual Languages and Computing*. 14: 341-362.

Wu, S., Liew, A.W.C., & Yan, H. (2004). Cluster analysis of gene expression data based on Self-Splitting and Merging Competitive Learning. *IEEE Transactions on Information Technology in Biomedicine*. 8(1): 5-15.

Yeung, K.Y., Fraley, C., Murua A., Raftery, A.E., & Ruzzo, W.L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*. 17(10): 977-987.

Zhang, Y.J., & Liu, Z.Q. (2002). Self-Splitting competitive learning: A new on-line clustering paradigm. *IEEE Transactions on Neural Networks*. 13(2): 369-380.

KEY TERMS

BHC Clustering: A clustering algorithm based on the hierarchical binary subdivision framework. The algorithm proceeds by partitioning a cluster into two classes, and then check for the validity of the partition by Fisher discriminant analysis. The algorithm terminates when no further valid subdivision is possible.

Biclustering: Also called two-way clustering or co-clustering. In biclustering, not only the objects but

also the features of the objects are clustered. If the data is represented in a data matrix, the rows and columns are clustered simultaneously.

Cluster Analysis: An exploratory data analysis technique that aims at finding groups in a data such that objects in the same group are similar to each other while objects in different groups are dissimilar.

DNA Microarray technology: A technology that allows massively parallel, high throughput genome-wide profiling of gene expression in a single hybridization experiment. The method is based on the complementary hybridization of DNA sequence.

Gene Expression: Gene expression is the process by which a gene's DNA sequence is converted into functional proteins. Some genes are turned on (expressed) or turned off (repressed) when there is a change in external conditions or stimuli.

Hierarchical Clustering: A clustering method that finds successive clusters using previously established clusters. Hierarchical algorithms can be agglomerative (bottom-up) or divisive (top-down). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

K-Means Clustering: K-means clustering is the most well-known partition-based clustering algorithm. The algorithm starts by choosing k initial centroids, usually at random. Then the algorithm alternates between updating the cluster assignment of each data point by associating with the closest centroid and updating the centroids based on the new clusters until convergence.

SSMCL Clustering: A partition-based clustering algorithm that is based on the one-prototype-take-one-cluster (OPTOC) learning paradigm. OPTOC learning is achieved by constructing a dynamic neighborhood that favors patterns inside the neighborhood. Eventually, the prototype settles at the center of a natural cluster, while ignoring competitions from other clusters. Together with the over-clustering and merging process, SSMCL is able to find all natural clusters in the data.

Clustering Algorithm for Arbitrary Data Sets

C

Yu-Chen Song
Inner Mongolia University of Science and Technology, China

Hai-Dong Meng
Inner Mongolia University of Science and Technology, China

INTRODUCTION

Clustering analysis is an intrinsic component of numerous applications, including pattern recognition, life sciences, image processing, web data analysis, earth sciences, and climate research. As an example, consider the biology domain. In any living cell that undergoes a biological process, different subsets of its genes are expressed in different stages of the process. To facilitate a deeper understanding of these processes, a clustering algorithm was developed (Ben-Dor, Shamir, & Yakhini, 1999) that enabled detailed analysis of gene expression data. Recent advances in proteomics technologies, such as two-hybrid, phage display and mass spectrometry, have enabled the creation of detailed maps of biomolecular interaction networks. To further understanding in this area, a clustering mechanism that detects densely connected regions in large protein-protein interaction networks that may represent molecular complexes was constructed (Bader & Hogue, 2003). In the interpretation of remote sensing images, clustering algorithms (Sander, Ester, Kriegel, & Xu, 1998) have been employed to recognize and understand the content of such images. In the management of web directories, document annotation is an important task. Given a predefined taxonomy, the objective is to identify a category related to the content of an unclassified document. Self-Organizing Maps have been harnessed to influence the learning process with knowledge encoded within a taxonomy (Adami, Avesani, & Sona, 2005). Earth scientists are interested in discovering areas of the ocean that have a demonstrable effect on climatic events on land, and the SNN clustering technique (Ertöz, Steinbach, & Kumar, 2002) is one example of a technique that has been adopted in this domain. Also, scientists have developed climate indices, which are time series that summarize the behavior of selected regions of the Earth's oceans and atmosphere. Clustering techniques have proved crucial in the production of climate indices (Steinbach, Tan, Kumar, Klooster, & Potter, 2003).

In many application domains, clusters of data are of arbitrary shape, size and density, and the number of clusters is unknown. In such scenarios, traditional clustering algorithms, including partitioning methods, hierarchical methods, density-based methods and grid-based methods, cannot identify clusters efficiently or accurately. Obviously, this is a critical limitation. In the following sections, a number of clustering methods are presented and discussed, after which the design of an algorithm based on Density and Density-reachable (CADD) is presented. CADD seeks to remedy some of the deficiencies of classical clustering approaches by robustly clustering data that is of arbitrary shape, size, and density in an effective and efficient manner.

BACKGROUND

Clustering aims to identify groups of objects (clusters) that satisfy some specific criteria, or share some common attribute. Clustering is a rich and diverse domain, and many concepts have been developed as the understanding of clustering develops and matures (Tan, Steinbach, & Kumar, 2006). As an example, consider spatial distribution. A typology of clusters based on this includes: Well-separated clusters, Center-based clusters, Contiguity-based clusters, and Density-based clusters. Given the diversity of domains in which clustering can be applied, and the diverse characteristic and requirements of each, it is not surprising that numerous clustering algorithms have been developed. The interested reader is referred to the academic literature (Qiu, Zhang, & Shen, 2005), (Ertöz, Steinbach, & Kumar, 2003), (Zhao, Song, Xie, & Song, 2003), (Ayad & Kamel, 2003), (Karypis, Han, & Kumar, 1999) for further information.

Though the range of clustering algorithms that have been developed is broad, it is posisble to classify them according the broad approach or method adopted by each:

- A partitioning method creates an initial set of k partitions, where the parameter k is the number of partitions to be constructed. Then it uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. Typical partitioning methods include K-means , K-medoids, CLARANS, and their derivatives.
- A hierarchical method creates a hierarchical decomposition of the given set of data objects. The method can be classified as being either agglomerative (bottom-up) or divisive (top-down), based on how the hierarchical decomposition is formed. To compensate for the rigidity of the merge or split, the quality of hierarchical agglomeration can be improved by analyzing object linkages at each hierarchical partition - an approach adopted by the CURE and Chameleon algorithms, or by integrating other clustering techniques such as iterative relocation – an approach adopted by BIRCH.
- A density-based method clusters objects based on the concept of density. It either grows the cluster according to the density of the neighborhood objects (an approach adopted by DBSCAN), or according to some density function (such that used by DENCLUE).
- A grid-based method first quantizes the object space into a finite number of cells thus forming a grid structure, and then performs clustering on the grid structure. STING is a typical example of a grid-based method based on statistical information stored in grid cells. CLIQUE and Wave Cluster are examples of two clustering algorithms that are both grid-based and density-based.
- A model-based method hypothesizes a model for each of the clusters and finds the best fit of the data to that model. Typical model-based methods involve statistical approaches (such as COBWER, CLASSIT, and AutoClass).

In essence, practically all clustering algorithms attempt to cluster data by trying to optimize some objective function.

DEVELOPMENT OF A CLUSTERING AGORITHM

Before the development of a clustering algorithm can be considered, it is necessary to consider some problems intrinsic to clustering. In data mining, efforts have focused on finding methods for efficient and effective cluster analysis in large databases. Active themes of research focus on the effectiveness of methods for clustering complex shapes and types of data, high-dimensional clustering techniques, scalability of clustering methods, and methods for clustering mixed numerical and categorical data in large databases. When clustering algorithms are analyzed, it is obvious that there are some intrinsic weaknesses that affect their applicability:

- Reliability to parameter selection - for partitioning methods and hierarchical methods, it is necessary to input parameters both for the number of clusters and the initial centroids of the clusters. This is difficult for unsupervised data mining when there is lack of relevant domain knowledge (Song & Meng, 2005 July), (Song & Meng, 2005 June). At the same time, different random initializations for number of clusters and centroids of clusters produce diverse clustering results, indicating a lack of stability on the part of the clustering method.
- Sensitivity to noise and outliers - noise and outliers can unduly influence the clusters derived by partitioning methods, hierarchical methods, grid-based methods, and model-based methods; however partitioning methods and hierarchical methods are particularly susceptible.
- Selectivity to cluster shapes - partitioning methods, hierarchical methods, and grid-based methods are not suitable for all types of data distribution, and cannot handle non-globular clusters of different shapes, sizes and densities.
- Ability to detect outliers - density-based methods are relatively resistant to noise and can handle clusters of arbitrary shapes and sizes, but can not detect outliers effectively.

Experimental Analysis of Traditional Clustering Agorithms

In order to visually illustrate the clustering results, two-dimensional data sets as experimental data are used. However, it should be noted that the analysis results are also suitable for higher dimensional data.

Partitioning Method

K-means, a partitioning method, is one of the most commonly used clustering algorithms, but it does not perform well on data with outliers or with clusters of different sizes or non-globular shapes. This clustering method is the most suitable for capturing clusters with globular shapes, but this approach is very sensitive to noise and cannot handle clusters of varying density.

Hierarchical Method

Hierarchical clustering techniques are a second important category of clustering methods, but the most commonly used methods are agglomerative hierarchical algorithms. Agglomerative hierarchical algorithms are expensive in terms of their computational and storage requirements. The space and time complexity of such algorithms severely restricts the size of the data sets that can be used. Agglomerative hierarchical algorithms identify globular clusters but cannot find non-globular clusters, and also cannot identify any outliers or noise points. CURE relies on an agglomerative hierarchical scheme to perform the actual clustering. The distance between two clusters is defined as the minimum distance between any two of their representative points (after the representative points are shrunk toward their

Figure 1. Clusters embedded in noise points of unequal density

respective centres). During this hierarchical clustering process, CURE eliminates outliers by eliminating small, slowly growing clusters. Although the concept of representative points does allow CURE to find clusters of different sizes and shapes in some data sets, CURE is still biased towards finding globular clusters, as it still incorporates the notion of a cluster centre.

Density-Based Methods

Density-based clustering locates regions of high density that are separated from one another by regions of low density. DBSCAN is a typical and effective density-based clustering algorithm which can find different types of clusters, identify outliers and noise, but it can not handle clusters of varying density. DBSCAN can have difficulty with density if the density of clusters and noise varies widely. Consider Figure 1, which illustrates three clusters embedded in noises of unequal densities. The noise around the clusters A and B has the same density as cluster C. If the Eps threshold is high enough such that DBSCAN finds A and B as separate clusters, and the points surrounding them are marked as noise, then C and the points surrounding it will also be marked as noise.

An Algorithm Based on Density and Density-Reachable

Based on the notions of density and density reachable (Meng & Zhang, 2006), (Meng & Song, 2005), a clustering algorithm that can find clusters of arbitrary shapes and sizes, handle clusters and noises of varying density, minimize the reliability of domain knowledge, and identify outliers efficiently, can be designed. A prerequisite to this is the provision of some key definitions:

1. The density of the data points is defined as the sum of the influence functions associated with each point:

$$density(x_i) = \sum_{j=1}^{n} e^{-\frac{d(x_i, x_j)^2}{2\sigma^2}}$$

where the Gaussian influence function $f_{Gauss}(x_i, x_j) = e^{-\frac{d(x_i, x_j)^2}{2\sigma^2}}$ indicates the density influence of each data points on the density of point x_i; and σ is the density adjustment parameter which

299
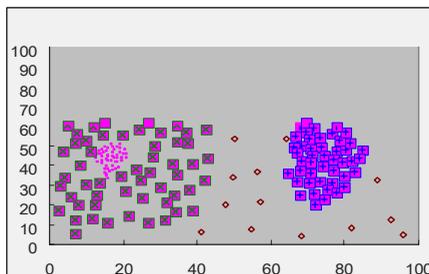
is analogous to be the standard deviation, and governs how quickly the influence of a point drops off.

2. Density-reachable distance is used to determine a circular area of data point x, labeled as $\delta = \{ \, x \mid 0 < d(x_i, x_j) \leq R \, \}$, the data points of which all belong the same cluster. The formula is:

$$R = \frac{mean(D)}{n^{coefR}}$$

where *mean(D)* is the mean distance between all data points in data set, and *coefR* is the adjustment coefficient of the density-reachable distance.

3. Local density attractors are the data points at which the value of the density function is a local maximum.

4. Density-reachable is defined as follows: if there is an object's chain $p_1, p_2, \ldots p_n, p_n = q$, q is a local density attractor, and p_{n-1} is density-reachable from q, then for $p_i \in D, (1 \leq i < n-1)$ and $d(p_i, p_{i+1}) < R$, we define that object $p_i, (1 \leq i < n-1)$ as being density-reachable from q.

THE CADD ALGORITHM

See Exhibit A.

Experimental Results

The clustering algorithm was developed using C++. In order to verify the effectiveness of the algorithm, a large number of experiments were carried out with different data sets which contain clusters of different shape, size, and density. The results are now considered.

Clusters of Complex Shapes

The clustering results of clusters of arbitrary shapes and sizes are shown in Figure 2 and Figure 3, the data set being taken from (Karypis et al., 1999).

In contrast to partitioning and hierarchical methods, the clustering algorithm CADD identified the clusters of arbitrary shapes and sizes, and identified the outliers effectively.

Clusters of Varying Density

Density-based methods can perform poorly when clusters have widely differing densities. The clustering algorithm CADD assign each object to a cluster accord-

Exhibit A.

Algorithm Clustering algorithm based on density and density-reachable（CADD）
Input：Data set, adjustment coefficient of density-reachable distance.
Output：Number of clusters, the members of each cluster, outliers or noise points.
Method：
1：Compute the densities of each data points and construct original data chain table of clustering objects.
2：i←1
3：**repeat**
4：Seek the maximum density attractor $O_{DensityMaxi}$ in the original data chain table of clustering objects as the first cluster center of C_i.
5：Assign the objects in the data chain which are density reachable from $O_{DensityMaxi}$ to cluster C_i,and at the same time delete the clustered objects form original data chain table.
6：i←i+1

C

Figure 2. Clustering result of winded clusters
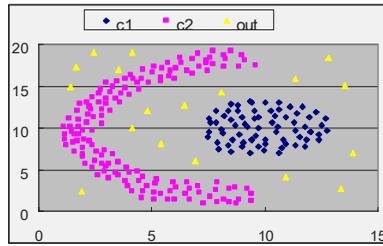
Figure 3. Clustering result of complex clusters
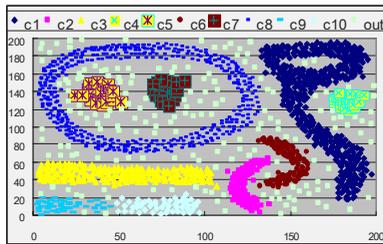
ing to density-reachable. When managing clusters of varying density, it is important to adjust the distance of density-reachable. This adjustment is carried out by multiplying the original density-reachable distance by the ratio of the density of the first density attractor to the density of the second density attractor, so as to succeed in identifying clusters of varying density. This is because when the density of the local density attractor of a cluster is larger, the distance between objects in the cluster is smaller, and conversely, when the density of the local density attractor of a cluster is smaller, the distance between objects in the cluster is larger.

Figure 4 demonstrates that CADD identifies three clusters C1, C2, and C3 of varying density and which are embedded in outliers or noise points of unequal density. The clustering result reflects the characteristics of data distribution, and is not affected by varying density.

Experimental Analysis of Computational Complexity

After increasing the size of the experimental data set, the run times of DBSCAN and CADD were recorded, as shown in Figure 5. The workstation used to perform the experiment was equipped with 1.6GHz CPU, 512MB

storage, and 80GB hard disk. All objects in the data set possessed 117 attributes.

The basic time complexity of DBSCAN is $O(n \times$ time to find objects in the *Eps*-neighborhood, where n is the number of data objects). In the worst case, this complexity is $O(n^2)$. However, in low-dimensional spaces, there are data structures, such as kd-trees, that allow efficient retrieval of all objects within a given distance of a specified point, and the time complexity can be as low as O(n log n). The space complexity of DBSCAN is $O(n)$. The basic time complexity of CADD is $O(kn)$, where k is the number of clusters. This is because it is only necessary to search the density-reachable objects in the data set once for each cluster, and in the worst case, the time complexity will be $O(n^2)$. The space requirement of CADD is $O(n)$.

FUTURE TRENDS

As many different clustering algorithms have been developed in a variety of domains for different types of applications, details of experimental clustering results

Figure 4. The result of clusters embedded in noises of unequal density
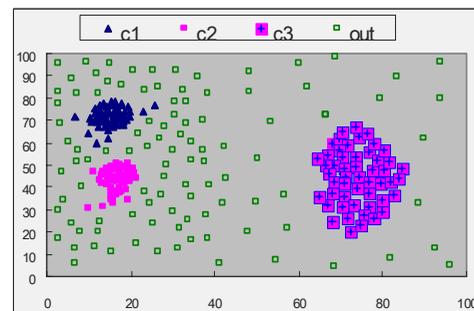
Figure 5. Run times of DBSCAN and CADD
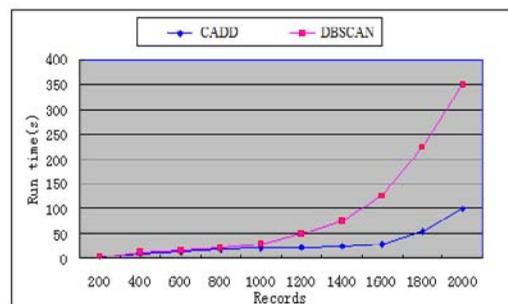
are discussed in many academic texts and research papers. None of these algorithms are suitable for every kind of data, clusters, and applications, and so there is significant scope for developing new clustering algorithms that are more efficient or better suited to a particular type of data, cluster, or application. This trend is likely to continue, as new sources of data are becoming increasingly available. Wireless Sensor Networks (WSNs) are an example of a new technology that can be deployed in a multitude of domains; and such networks will almost invariably give rise to significant quantities of data that will require sophisticated analysis if meaningful interpretations are to be attained.

CONCLUSION

Clustering forms an important component in many applications, and the need for sophisticated, robust clustering algorithms is likely to increase over time. One example of such an algorithm is CADD. Based on the concepts of density and density-reachable, it is overcomes some of the intrinsic limitations of traditional clustering mechanisms, and its improved computational efficiency and scalability have been verified experimentally.

ACKNOWLEDGMENT

REFERENCES

Adami, G., Avesani, P., & Sona, D. (2005). Clustering documents into a web directory for bootstrapping a supervised classification. Data & Knowledge Engineering, 54, 301-325.

Ayad, H., Kamel, M. (2003). Finding Natural Clusters Using Multi-clusterer Combiner Based on Shared Nearest Neighbors. Springer Berlin, 2709, 159-175.

Bader, G. D. & Hogue, C. W. (2003). An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics, 4. http://www.biomedcentral.com/1471-2105/4/2

Ben-Dor, A., Shamir, R., & Yakhini, Z. (1999). Clustering gene expression patterns. Journal of computational Biology, 6, 281-297.

Ertöz, L., Steinbach, M., & Kumar, V. (2003). Finding Clusters of Different Size, Shapes and Densities in Noisy High Dimensional Data. In Proc. of the 3rd SIAM International Conference on Data mining.

Ertöz, L., Steinbach, M., & Kumar, V. (2002). A New Shared Nearest Neighbor Clustering Algorithm and its Applications. In: Proceedings of the Work-shop on Clustering High Dimensional Data and its Applications, Second SIAM International Conference on data mining, Arlington, VA, USA.

Karypis, G., Han, E-H., & Kumar, V. (1999), CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Computer 32(8), 68-75.

Meng, H-D., & Zhang, Y-Y. (2006). Improved Clustering Algorithm Based on Density and Direction. Computer Engineering and Applications, 42(20), 154-156.

Meng, H-D., & Song, Y-C. (2005). The implementation and application of data mining system based on campus network. Journal on communications, 26(1A), 185-187.

Qiu, B.Z., Zhang, X-Z., & Shen, J-Y. (2005). Grid-based clustering algorithm for multi-density. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 3, 1509-1512.

Sander, J. O., Ester, M., Kriegel, H-P., & Xu, X. W. (1998). Density-Based Clustering in Spatial Data sets: The Algorithm GDBSCAN and Its Applications. Data mining and Knowledge Discovery, 2, 169–194.

Song, Y-C., & Meng, H-D. (2005, July). The design of expert system of market basket analysis based on data mining. Market modernization, 7,184-185.

Song, Y-C., & Meng, H-D. (2005, June). The construction of knowledge base based on data mining for market basket analysis. Market modernization, 6,152-153.

Steinbach, M., Tan, P-N., Kumar, V., Klooster, S., & Potter, C. (2003) Discovery of climate indices using clustering. Proceedings of the ninth ACM SIGKDD

international conference on Knowledge discovery and data mining, Washington, D.C., 446--455

Tan, P-N., Steinbach, M., & Kumar, V. (2006). Introduction to Data mining. Post & Telecom Press of P.R.China (Chinese Edition), 359-371.

Zhao, Y-C., Song, M., Xie, F., & Song, J-D. (2003). Clustering Datasets Containing Clusters of Various Densities. Journal of Beijing University of Posts and Telecommunications, 26, 42-47.

KEY TERMS

Cluster Analysis: Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships.

CADD: A clustering algorithm based on the concepts of Density and Density-reachable.

Centre-Based Clusters: Each object in a centre-based cluster is closer to the centre of the cluster than to the centres of any other clusters.

Contiguity-Based Clusters: Each object in a contiguity-based cluster is closer to some other object in the cluster than to any point in a different cluster.

Density-Based Clusters: Each object in a density-based cluster is closer to some other object within its Eps neighbourhood than to any object not in the cluster, resulting in dense regions of objects being surrounded by regions of lower density.

Eps: Maximum radius of the neighbourhood.

Well-Separated Cluster: A cluster is a set of objects in which each object is significantly closer (or more similar) to every other object in the cluster than to any object not in the cluster.

CNS Tumor Prediction Using Gene Expression Data Part I

Atiq Islam
University of Memphis, USA

Khan M. Iftekharuddin
University of Memphis, USA

E. Olusegun George
University of Memphis, USA

David J. Russomanno
University of Memphis, USA

INTRODUCTION

Automated diagnosis and prognosis of tumors of the central nervous system (CNS) offer overwhelming challenges because of heterogeneous phenotype and genotype behavior of tumor cells (Yang et al. 2003, Pomeroy et al. 2002). Unambiguous characterization of these tumors is essential for accurate prognosis and therapy. Although the present imaging techniques help to explore the anatomical features of brain tumors, they do not provide an effective means of early detection. Currently, the histological examination of brain tumors is widely used for an accurate diagnosis; however, the tumor classification and grading based on histological appearance does not always guarantee absolute accuracy (Yang et al., 2003, Pomeroy et al., 2002). In many cases, it may not be sufficient to detect the detailed changes in the molecular level using a histological examination (Yang et al. 2003) since such examination may not allow accurate prediction of therapeutic responses or prognosis. If the biopsy sample is too small, the problems are aggravated further.

Toward achieving a more reliable diagnosis and prognosis of brain tumors, gene expression measures from microarrays are the center of attention to many researchers who are working on tumor prediction schemes. Our proposed tumor prediction scheme is discussed in two chapters in this volume. In part I (this chapter), we use an analysis of variance (ANOVA) model for characterizing the Affymetrix gene expression data from CNS tumor samples (Pomeroy et al. 2002) while in part II we discuss the prediction of tumor classes based on marker genes selected using the techniques developed in this chapter. In this chapter, we estimate the tumor-specific gene expression measures based on the ANOVA model and exploit them to locate the significantly differentially expressed marker genes among different types of tumor samples. We also provide a novel visualization method to validate the marker gene selection process.

BACKGROUND

Numerous statistical methods have evolved that are focused on the problem of finding the marker genes that are differentially expressed among tumor samples (Pomeroy et al., 2002, Islam et al., 2005, Dettling et al., 2002, Boom et al., 2003, Park et al., 2001). For example, Pomeroy et al. (2002) uses student t-test to identify such genes in embryonal CNS tumor samples. Because of the non-normality of gene expression measurements, several investigators have adopted the use of nonparametric methods, such as the Wilcoxon Sum Rank Test (Wilcoxon, 1945) as a robust alternative to the parametric procedures. In this chapter, we investigate a Wilcoxon-type approach and adapt the resulting procedures for locating marker genes.

Typically, statistical procedures for microarray data analysis involve performing gene specific tests. Since the number of genes under consideration is usually large, it is common practice to control the potentially large number of false-positive conclusions and family-wise error rates (the probability of at least one

C

false positive statement) through the use of P-value adjustments. Pollard et al. (2003) and Van der Laan et al. (2004a, 2004b, 2005c) proposed methods to control family-wise error rates based on the bootstrap resampling technique of Westfall & Young (1993). Benjamini & Hochberg (1995), Efron et al. (2001) and Storey et al. (2002, 2003a, 2003b, 2004) introduced various techniques for controlling the false discovery rate (FDR), which is defined as the expected rate of falsely rejecting the null hypotheses of no differential gene expression. These adjustment techniques have gained prominence in statistical research relating to microarray data analysis. Here, we use FDR control because it is less conservative than family-wise error rates for adjusting the observed P-values for false discovery. In addition, we propose a novel marker gene visualization technique to explore appropriate cutoff selection in the marker gene selection process.

Before performing formal analysis, one should identify the actual gene expression levels associated with different tissue groups and discard or minimize other sources of variations. Such an approach has been proposed by Townsend & Hartl (2002) who use a Bayesian model with both multiplicative and additive small error terms to detect small, but significant differences in gene expressions. As an alternative, an ANOVA model appears to be a natural choice for estimating true gene expression (Kerr et al., 2000, Pavlidis et al., 2001, Wolfinger et al. 2001). In the context of cDNA microarray data, the ANOVA model was first proposed by Kerr et al. (2000).

TUMOR-SPECIFIC GENE EXPRESSION ESTIMATION AND VISUALIZATION TECHNIQUES

To illustrate our procedure, we use the a microarray data set by Pomeroy et al. (2002) of patients with different types of embryonal tumors. The patients include 60 children with medulloblastomas, 10 young adults with malignant gliomas, 5 children with AT/RTs, 5 with renal/extra-renal rhabdoid tumors, and 8 children with supratentorial PNETs. First, we preprocess the data to remove extraneous background noise and array effects. To facilitate our analysis, we divide the dataset into groups as shown in Fig. 1. We rescale the raw expression data obtained from Affymetrix's GeneChip to account for different chip intensities.

Microarray data typically suffer from unwanted sources of variation, such as large-and-small-scale intensity fluctuations within spots, non-additive background, fabrication artifacts, probe coupling and processing procedures, target and array preparation in the hybridization process, background and over-shining effects, and scanner settings (McLachlan & Ambroise, 2005). To model these variations, a number of methods have been reported in the literature (Kerr et al., 2000, Lee et al., 2000, Pavlidis, 2001, Wolfinger et al., 2001, Ranz et al., 2003, Townsend, 2004, Tadesse et al., 2005). An ANOVA model similar to the one used by Kerr el al. (2000) is adopted in our current work and facilitates obtaining the tumor-specific gene expression measures from the preprocessed microarray data. Our two-way ANOVA model is given as:

$$y_{jgk} = \mu + \alpha_g + \beta_j + \gamma_{jg} + \varepsilon_{jgk} \qquad (1)$$

Figure 1. Dataset grouping
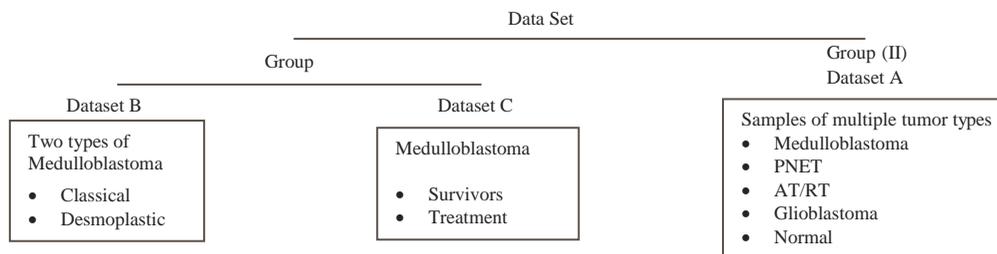
where, y_{jgk} denotes the log of the gene expression measure for the *k*th replicate of the *g*th gene in the *j*th tumor group (k = 1,…, K_j; g = 1,…, G; j = 1,…, J); μ, $α_g$, $β_j$, $γ_{jg}$ refer to the overall effect, the *g*th gene main effect, the *j*th tumor-group main effect, and the *g*th gene - *j*th tumor group interaction effect, respectively, and $ε_{jgk}$ is a random error with zero mean and constant variance. We assume that these terms are independent; however, we do not make any assumption about their distribution.

This model assumes that background noise and array effects have been eliminated at previous preprocessing steps. This assumption fits well with our preprocessed data. A reason for selecting the model is that it fits well with our goal to build the suitable tumor prototypes for prediction. We believe that tumor prototypes should be built based only on the tumor-specific gene expression measures. In this model, the interaction term $γ_{jg}$ constitutes the actual gene expression of gene g attributed to the tumor type j (McLachlan & Ambroise, 2005). Hence, the value of the contribution to the tumor specific gene expression value by the *k*th replication of the measurement on gene g in tissue j may be written as given in (McLachlan & Ambroise, 2005), as:

$$\hat{γ}_{jgk} = y_{jgk} - \hat{μ} - \hat{α}_g - \hat{β}_j \qquad (2)$$

and the tumor specific expression is estimated by Equation (3):

$$\overline{γ}_{jg} = \frac{1}{K_j} \sum_{k=1}^{K_j} \hat{γ}_{jgk} \qquad (3)$$

where, $\hat{μ} = \overline{y}_{...}$, $\hat{α}_g = \overline{y}_{.g.} - \overline{y}_{...}$ and $\hat{β}_j = \overline{y}_{j..} - \overline{y}_{...}$ are the least square estimates of the *g*th gene and *j*th tumor-group main effects based on replications (Lee et al., 2000, McLachlan & Ambroise, 2005). Here, we regard these estimates as fixed-effects estimates. In a Bayesian analysis, all of the parameters could be treated as random effects. The $\hat{γ}_{jgk}$ values are considered in our subsequent steps of the contribution of replicate k to the gene expression of gene g for a patient in the j-th tumor group.

Using several different procedures, such as Shapiro-Wilk's test (Shapiro & Wilk, 1965) and normal probability plots, we observe that the gene expression levels in our dataset are not normally distributed. Hence, we choose a nonparametric test, such as Wilcoxon

(Wilcoxon, 1945), for the two categories problem and the Kruskal-Wallis (NIST, 2007) for the five categories problem in our dataset to identify significantly differentially expressed genes among the tissue sample types. We adjust for multiplicity of the tests involved by controlling the False Discovery Rate (FDR) using q-values as proposed by Storey et al. (2004).

In the selection of differentially expressed genes, a tight cutoff may miss some of the important marker genes, while a generous threshold increases the number of false positives. To address this issue, we use the parallel coordinate plot (Inselberg et al., 1990) of the group-wise average genes expressions. In this plot, the parallel and equally spaced axes represent individual genes. Separate polylines are drawn for each group of tumor samples. The more the average gene expression levels differ between groups, the more space appears among the polylines in the plot. To effectively visualize the differentially expressed genes, we first obtain the average of the tumor-specific gene expression values within any specific tissue sample type $\overline{γ}_{jg}$ as specified in Equation (3). We then standardize the average gene expression values $\overline{γ}_{jg}$ obtained in Equation (3) as follows:

$$\widehat{γ}_{jg} = \frac{\overline{γ}_{jg}}{S_g} \qquad (4)$$

where

$$S_g^2 = \frac{\sum_j (K_j - 1) S_{jg}^2}{\sum_j (K_j - 1)} \qquad (5)$$

and

$$S_{jg}^2 = \frac{\sum_{k=1}^{K_j} (γ_{jgk} - \overline{γ}_{jg})^2}{K_j - 1}. \qquad (6)$$

Next, we divide the genes into two groups. The first group consists of the genes where $\widehat{γ}_{1g} \geq \widehat{γ}_{2g}$ and the remainder of the genes are kept in the second group. We group such that the tumor-type representing lines in our plot do not cross. Now, within each gene group, we again partition the genes into subgroups so that similarly expressed genes are grouped together. The

C

Self-Organizing Map (SOM) (Kohonen, 1987) analysis of variance approach is exploited for this partitioning. Then, within each of the subgroups, genes are ordered according to $\left|\widehat{\gamma}_{1g} - \widehat{\gamma}_{2g}\right|$. This further partitioning and ordering method confers suitable shapes to the tumor-type representing lines such that the user can quickly visualize the tumor-type discriminating power of the selected genes. Before generating the final plot we normalize the standardized average expression values $\widehat{\gamma}_{jg}$ as follows:

$$\widetilde{\gamma}_{jg} = \frac{\widehat{\gamma}_{jg} - \min_{j,g}(\widehat{\gamma}_{jg})}{\max_{j,g}(\widehat{\gamma}_{jg}) - \min_{j,g}(\widehat{\gamma}_{jg})} \qquad (7)$$

Finally, the normalized expression values $\widetilde{\gamma}_{jg}$ are plotted using parallel coordinates, where each parallel axis corresponds to a specific gene and each polyline corresponds to a specific tumor type. Each gene's subgroups are plotted in separate plots. Algorithm 1 specifies our formalized approach to gene visualization. The purpose of such plots is to qualitatively measure the performance of the gene selection process and to find the appropriate cutoff that reduces the number of false positives while keeping the number of true positives high.

The following results illustrate the usefulness of our visualization method provided in Algorithm 1. The expression patterns of the marker genes associated with medulloblastoma survivor and failure groups is shown in Figs. 2 and 3. The solid and dotted lines represent the failure (death) and survivor (alive) groups, respectively. Genes are selected using the Wilcoxon method, which was previously described, wherein depending on the q-values, different numbers of genes are selected. In both Figs. 2 and 3, each individual graph represents a group of similarly expressed genes clustered together as specified in step 5 of Algorithm 1. Figs. 2 and 3 show 280 and 54 selected marker genes, respectively. We observe that within 280 selected genes in Fig. 2, many show similar expression patterns in both failure and survivor sample groups indicating that two sample groups are close to one another on the parallel axes. Since each axis represents a different gene, we conclude that many of the genes in Fig. 2 are falsely identified as marker genes (false positives). In comparison, the solid and dotted lines are far apart on most of the parallel axes in Fig. 3. This indicates that the average gene expression values of the selected 54 genes are quite different between the two groups of tissue samples. Thus, this visualization aids in selecting the correct threshold in the marker gene selection process.

Algorithm 1 (*DATA*) to visualize the expression pattern of the selected marker genes

DATA contains the expression levels of the marker genes for all the patient samples; where each row represents different gene expression values and each column represents different patient samples.

1. Sort the genes in descending order according to the values of q-values and select the top G genes from the sorted list.

2. Estimate the average of the tumor-specific gene expression values $\overline{\gamma}_{jg}$ within any specific tissue sample type using Eq. (3)

3. Obtain the standardized average gene expression values $\widehat{\gamma}_{jg}$ using Eq. (4).

4. Partition the genes into two groups: (i) C_1 where $\widehat{\gamma}_{1g} \geq \widehat{\gamma}_{2g}$ and (ii) C_2 where $\widehat{\gamma}_{1g} < \widehat{\gamma}_{2g}$.

5. Within each group C_c (c=1, 2), again partition the gene expression values $\widehat{\gamma}_{jg}$ into P clusters $\{C_{c1}... C_{cp}\}$ (c=1, 2) exploiting SOM, where each cluster consists of a group of similarly expressed genes.

6. For each of the clusters obtained in the previous step:

   a. Order the genes according to $\left|\widehat{\gamma}_{1g} - \widehat{\gamma}_{2g}\right|$.

   b. Obtain $\widetilde{\gamma}_{jg}$ by normalizing the standardized average expression values $\widehat{\gamma}_{jg}$ using Eq. (7).

   c. Plot the normalized average expression values $\widetilde{\gamma}_{jg}$ using ==parallel coordinates==, where each parallel axes corresponds to a specific gene and each polyline corresponds to a specific tissue sample type.

Figure 2. Expression patterns of the marker genes associated with medulloblastoma survivor and failure groups. Genes are selected using Wilcoxon method and FDR is controlled using q-values, where depending on the q-values, different numbers of genes are selected 280 genes.
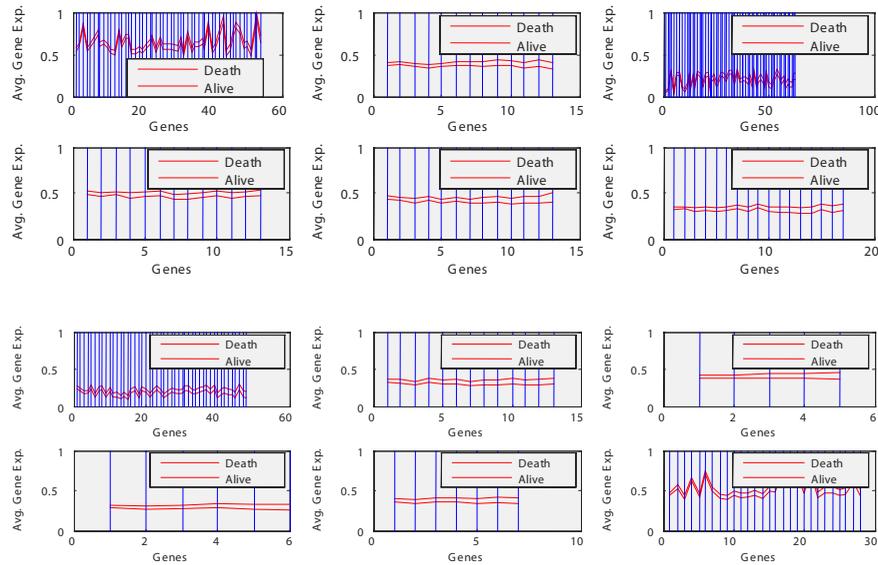
Figure 3. Expression patterns of the marker genes associated with medulloblastoma survivor and failure groups. Genes are selected using Wilcoxon method and FDR is controlled using q-values, where depending on the q-values, different numbers of genes are selected 54 genes.
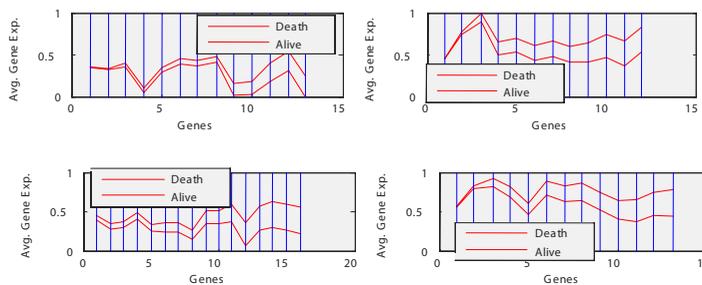
FUTURE TRENDS

In this chapter, we estimate the tumor-specific gene expression measure exploiting an ANOVA model. We specify the model parameters as fixed effects; however, such specification may not be always appropriate. Rather, considering the model parameters as random effects may be more appropriate for microarray dataset (Kerr et al., 2000). Thus, one possible improvement may consider all of the effects in our ANOVA model as random. Further, specifying the random effect parameters in a Bayesian framework provides a formal way of exploiting any prior knowledge about the parameter distribution, if available. We are currently in the process

C

of adopting a hierarchical Bayesian approach to our work following a few more recent relevant works by (Ibrahim et al., 2002, Lewin et al., 2006).

CONCLUSION

We attempted to estimate tumor-specific gene expression measures using an ANOVA model. These estimates are then used to identify differentially expressed marker genes. For evaluating the marker gene identification, we proposed a novel approach to visualize the average genes expression values for a specific tissue type. The proposed visualization plot is useful to qualitatively evaluate the performance of marker gene selection methods, as well as to locate the appropriate cutoffs in the selection process. The research in this chapter was supported in part through research grants [RG-01-0125, TG-04-0026] provided by the Whitaker Foundation with Khan M. Iftekharuddin as the principal investigator.

REFERENCES

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society,* Series B, 57(1), 289-300.

Boom, J., Wolter, M., Kuick, R., Misek, D.E., Youkilis, A.S., Wechsler, D.S., Sommer, C., Reifenberger, G., & Hanash, S.M. (2003). Characterization of Gene Expression Profiles Associated with Glioma Progression Using Oligonucleotide-Based Microarray Analysis and Real-Time Reverse Transcription-Polymerase Chain Reaction. *American Journal of Pathology*, 163(3), 1033-1043.

Dettling, M., & Buhlmann, P. (2002). Supervised Clustering of Genes. *Genome Biology*, 3(12), 1-15.

Efron, B., Tibshirani, R., Storey, J.D., & Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment. *Journal of American Statistics Association*, 96(115), 1–60.

Ibrahim, J.G., Chen, M.H., & Gray, R.J. (2002). Bayesian Models for Gene Expression with DNA Microarray Data. *Journal of the American Statistical Association, Applications and Case Studies*, 97(457), 88-99.

Inselberg, A. & Dimsdale, B. (1990). Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. *In Proceedings of IEEE Conference on Visualization*, 361-378.

Islam, A., & Iftekharuddin, K. (2005). Identification of Differentially Expressed Genes for CNS Tumor Groups and Their Utilization in Prognosis. *Memphis Bio-Imaging Symposium*.

Kerr, Martin, Churchill. (2000). Analysis Of Variance for Gene Expression Microarray Data. *Journal of Computational Biology*, 7, 819-837.

Kohonen, T. (1987). Self-Organization and Associative Memory. *Springer-Verlag*, 2nd Edition.

Lee, M.L.T., Kuo, F.C., Whitmore, G.A., & Sklar, J. (2000). Importance of Replication in Microarray Gene Expression Studies: Statistical Methods and Evidence from Repetitive cDNA Hybridizations. *Proceedings of the National Academy of Sciences,* 97, 9834-9838.

Lewin, A., Richardson, S., Marshall, C., Glazier, A., & Aitman, T. (2006). Bayesian Modeling of Differential Gene Expression. *Biometrics*, 62(1), 10-18.

McLachlan, G.J. & Ambroise, K.D. (2005). Analyzing Microarray Gene Expression Data. *John Wiley*.

NIST/SEMATECH. (2007). e-Handbook of Statistical Methods. *http://www.itl.nist.gov/div898/handbook*.

Park, P., Pagano, M., & Bonetti, M. (2001). A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data. *Pacific Symposium on Biocomputing*, 6, 52-53.

Pavlidis, P., & Noble, W.S. (2001). Analysis of Strain and Regional Variation of Gene Expression in Mouse Brain. *Genome Biology*, 2, 0042.1-0042.15.

Pollard, K.S., & Van der Laan, M.J. (2005). Resampling-based Multiple Testing: Asymptotic Control of Type I Error and Applications to Gene Expression Data. *Journal of Statistical Planning and Inference*, 125, 85-100.

Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., Sturla, L.M., Angelo, M., McLaughlin, M.E., Kim, J.Y., Goumnerova, L.C., Black, P.M., Lau, C., Allen, J.C., Zagzag, D., Olson, J.M., Curran, T., Wetmore, C., Biegel, J.A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D.N., Mesirov,

J.P., Lander, E.S., & Golub, T.R. (2002). Prediction of Central Nervous System Embryonal Tumor Outcome Based on Gene Expression. *Nature*, 415, 436-442.

Ranz, J.M., Castillo-Davis, C.I., Meiklejohn, C.D., & Hartl, D.L. (2003). Sex-Dependent Gene Expression and Evolution of the Drosophila Transcription. *Science*, 300, 1742-1745.

Shapiro, S.S. & Wilk. M.B. (1965). An Analysis of Variance Test for Normality (Complete Samples). *Biometrika*, 52(3), 591-611.

Storey, J.D. (2002). A Direct Approach to False Discovery Rates. *Journal of the Royal Statistical Society*, Series B, 64, 479-498.

Storey, J.D., & Tibshirani, R. (2003a). Statistical Significance for Genome-Wide Experiments. *Proceedings of the National Academy of Sciences*, 100, 9440-9445.

Storey, J.D. (2003b). The Positive False Discovery Rate: A Bayesian Interpretation and the Q-Value. *Annals of Statistics*, 31, 2013-2035.

Storey, J.D., Taylor, J.E., & Siegmund, D. (2004). Strong Control, Conservative Point Estimation, and Simultaneous Conservative Consistency of False Discovery Rates: A Unified Approach. *Journal of the Royal Statistical Society*, Series B, 66, 187-205.

Tadesse, M.G., Ibrahim, J.G., Gentleman, R., & Chiaretti, S. (2005). Bayesian Error-In-Variable Survival Model for the Analysis of Genechip Arrays. *Biometrics*, 61, 488-497.

Townsend, J.P., & Hartl, D.L. (2002). Bayesian Analysis of Gene Expression Levels: Statistical Quantification of Relative mRNA Level across Multiple Strains or Treatments. *Genome Biology*, 3, 1-71.

Townsend, J.P. (2003). Multifactorial Experimental Design and the Transitivity of Ratios with Spotted DNA Microarrays. *BMC Genomics*, 4, 41.

Townsend J.P. (2004). Resolution of Large and Small Differences in Gene Expression Using Models for the Bayesian Analysis of Gene Expression Levels and Spotted DNA Microarrays. *BMC Bioinformatics*, 5, 54.

Van der Laan, M.J., Dudoit, S., & Pollard, K.S. (2004a). Augmentation Procedures for Control of the Generalized Family-Wise Error Rate and Tail Probabilities for the Proportion of False Positives. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 15.

Van der Laan, M.J., Dudoit, S., & Pollard, K.S. (2004b). Multiple Testing. Part I. Single-Step Procedures for Control of General Type I Error Rates. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 13.

Van der Laan, M.J., Dudoit, S., & Pollard, K.S. (2004c). Multiple Testing. Part II. Step-Down Procedures for Control of the Family-Wise Error Rate. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 14.

Westfall, P.H., & Young, S.S. (1993). Resampling Based Multiple Testing: Examples and Methods for P-Value Adjustment. *New York: Wiley*.

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics*, 1, 80-83.

Wolfinger, R.D., Gibson, G., Wolfinger, E .D., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C., & Paules, R.S. (2001). Assessing Gene Significance from cDNA Microarray Expression Data via Mixed Models. *Journal of Computational Biology*, 8, 625-637.

Yang, Y., Guccione, S., & Bednarski, M.D. (2003). Comparing Genomic and Histologic Correlations to Radiographic Changes in Tumors: A Murine SCC Vll Model Study. *Academic Radiology*, 10(10), 1165-1175.

KEY TERMS

DNA Microarray: A collection of microscopic DNA spots, commonly representing single genes, arrayed on a solid surface by covalent attachment to chemically suitable matrices.

False Discovery Rate (FDR): Controls the expected proportion of false positives instead of controlling the chance of any false positives. An FDR threshold is determined from the observed p-value distribution from multiple single hypothesis tests.

Histologic Examination: The examination of tissue specimens under a microscope.

Kruskal-Wallis Test: A nonparametric mean test which can be applied if the number of sample groups is more than two, unlike the Wilcoxon Rank Sum Test.

Parallel Coordinates: A data visualization scheme that exploits 2D pattern recognition capabilities of humans. In this plot, the axes are equally spaced and are arranged parallel to one another rather than being arranged mutually perpendicular as in the Cartesian scenario.

q-values: A means to measure the proportion of FDR when any particular test is called significant.

Wilcoxon Rank Sum Test: A nonparametric alternative to the two sample t-test which is based on the order in which the observations from the two samples fall**.**

CNS Tumor Prediction Using Gene Expression Data Part II

Atiq Islam
University of Memphis, USA

Khan M. Iftekharuddin
University of Memphis, USA

E. Olusegun George
University of Memphis, USA

David J. Russomanno
University of Memphis, USA

INTRODUCTION

In this chapter, we propose a novel algorithm for characterizing a variety of CNS tumors. The proposed algorithm is illustrated with an analysis of an Affymetrix gene expression data from CNS tumor samples (Pomeroy et al., 2002). As discussed in the previous chapter entitled: CNS Tumor Prediction Using Gene Expression Data Part I, we used an ANOVA model to normalize the microarray gene expression measurements. In this chapter, we introduce a systemic way of building tumor prototypes to facilitate automatic prediction of CNS tumors.

BACKGROUND

DNA microarrays, also known as genome or DNA chips, have become an important tool for predicting CNS tumor types (Pomeroy et al., 2002, Islam et al., 2005, Dettling et al., 2002). Several researchers have shown that cluster analysis of DNA microarray gene expression data is helpful in finding the functionally similar genes and also to predict different cancer types. Eisen et al. (1998) used average linkage hierarchical clustering with correlation coefficient as the similarity measure in organizing gene expression values from microarray data. They showed that functionally similar genes group into the same cluster. Herwig et al. (1999) proposed a variant of the K-means algorithm to cluster genes of cDNA clones. Tomayo et al. (1999) used self-organized feature maps (SOFMs) to organize genes into biologically relevant groups. They found that SOFMs reveal true cluster structure compared to the rigid structure of hierarchical clustering and the structureless K-means approach. Considering the many-to-many relationships between genes and their functions, Dembele et al. (2003) proposed a fuzzy C-means clustering technique. The central goal of these clustering procedures (Eisen et al., 1998, Herwig et al., 1999, Tomayo et al., 1999, Dembele et al., 2003) was to group genes based on their functionality. However, none of these works provide any systematic way of discovering or predicting tissue sample groups as we propose in our current work.

To identify tissue sample groups, Alon et al. (1999) proposed a clustering algorithm that uses a deterministic-annealing algorithm to organize the data in a binary tree. Alizadeh et al. (2000) demonstrated a successful molecular classification scheme for cancers from gene expression patterns by using an average linkage hierarchical clustering algorithm with Pearson's correlation as the similarity measure. However, no formal way of predicting the category of a new tissue sample is reported in (Alon et al., 1999, Alizadeh et al., 2000). Such class prediction problems were addressed by Golub et al. (1999) who used SOFMs to successfully discriminate between two types of human acute leukemia. Dettling et al. (2002) incorporated the response variables into gene clustering and located differentially expressed groups of genes from the clustering result. These gene groups were then used to predict the categories of new samples. However, none of the above-mentioned works (Dettling et al., 2002, Golub et al., 1999, Alon et al.,

C

1999, Alizadeh et al., 2000) considered the correlation among the genes in classifying and/or predicting tissue samples. Moreover, none of these provided any systematic way of handling the probable subgroups within the known groups. In this chapter, we consider both correlations among the genes and probable subgroups within the known groups by forming appropriate tumor prototypes. Further, a major drawback of these analyses (Dettling et al., 2002, Eisen et al., 1998, Herwig et al., 1999, Tomayo et al., 1999, Dembele et al., 2003, Golub et al., 1999, Alon et al., 1999, Alizadeh et al., 2000) is insufficient normalization. Although, most of these methods normalize the dataset to remove the array effects; they do not concentrate on removing other sources of variations present in the microarray data.

Our primary objective in this chapter is to develop an automated prediction scheme for CNS tumors, based on DNA microarray gene expressions of tissue samples. We propose a novel algorithm for deriving prototypes for different CNS tumor types, based on Affymetrix HuGeneFL microarray gene expression data from Pomeroy et al. (2002). In classifying the CNS tumor samples based on gene expression, we consider molecular information, such as the correlations among gene expressions and probable subgroupings within the known histological tumor types. We demonstrate how the model can be utilized in CNS tumor prediction.

CNS TUMOR PROTOTYPE FOR AUTOMATIC TUMOR DETECTION

The workflow to build the tumor prototypes is shown in Fig. 1. In the first step, we obtain the tumor-type-specific gene expression measures. Then, we identify the marker genes that are significantly differentially expressed among tissue types. Next, a visualization technique is used to analyze the appropriateness of the marker gene selection process. We organize the marker genes in groups so that highly correlated genes are grouped together. In this clustering process, genes are grouped based on their tumor-type-specific gene expression measures. Then, we obtain eigengene expressions measures from each individual gene group by projection of gene expressions into the first few principal components. At the end of this step, we replace the gene expression measurements with eigengene expression values that conserve correlations between strongly correlated genes. We then divide the tissue samples of known tumor types into subgroups. The centroids of these subgroups of tissue samples with eigengene expressions represent the prototype of the corresponding tumor type. Finally, any new tissue sample is predicted as the tumor type of the closest centroid. This proposed novel prediction scheme considers both the correlation among the highly correlated genes and the probable phenotypic subgrouping within the known tumor types. These issues are often ignored in the literature for predicting tumor categories. The detail of the steps up to the identification of marker genes are provided in the previous chapter entitled: CNS Tumor Prediction Using Gene Expression Data Part I. In this section, we provide the details of the subsequent steps.

Now, we discuss the creation of the tumor prototypes using the tumor-specific expression values of our significantly differentially expressed marker genes identified in the previous step. Many of the marker genes are likely to be highly correlated. Such correlations of the genes affect successful tumor classification. However, this gene-to-gene correlation may provide important biological information. Hence, the inclusion of the appropriate gene-to-gene correlations in the tumor model may help to obtain a more biologically meaningful tumor prediction. To address this non-trivial need, we first group the highly correlated genes using the complete linkage hierarchical approach wherein correlation coefficient is considered as the pair-wise similarity measure of the genes. Next, for each of the clusters, we compute the principal components (PCs) and project the genes of the corresponding cluster onto the first 3 PCs to obtain eigengene expressions (Speed, 2003). Note that the PCs and the eigengene expressions are computed separately for each cluster. Such eigengenes encode the correlation information among

Figure 1. Simplified workflow to build the tumor prototypes

the highly correlated genes that are clustered together. Recently, molecularly distinct sub-grouping within the same histological tumor type has been reported (Taylor et al., 2005). To find a subgrouping within the same histological tumor type, we again use self-organizing maps (SOMs) (Kohonen, 1987) to cluster the tissue samples within each tumor group. This subgrouping within each group captures the possible genotypic variations within the same histological tumor type. Now, the prototype of any specific histological tumor type is composed of the centroid obtained from the corresponding SOM grid. Algorithm 1 shows our steps for building the tumor prototype.

To predict the tumor category of any new sample, we calculate the distances between the new sample and each of the prototype subgroups obtained using Algorithm 1. The category of the sample is predicted as that of the closest subgroup. The distance between the new sample and the xth subgroup, d_x, is calculated based on Euclidean distance as follows:

$$d_x = \sqrt{\sum_{k=1}^{N} (\overline{g}_k - g_k)^2} \qquad (1)$$

where $\overline{g}_{xk}$ is the center value of kth eigengene in the xth subgroup, g_k is the expression measure of kth eigengene of the new sample, and N is the total number of eigengenes. This distance measure deliberately ignores the non-representative correlations among the eigengene expressions since they are not natural and hence difficult to interpret.

Table 1 shows the efficacy of our model in classifying five categories of tissues simultaneously. Table 2 shows the performance comparison between our proposed prediction scheme and the method adopted in (Pomeroy et al., 2002). We observe that our prediction scheme outperforms the other prediction method in all three cases. The most noticeable difference is with data group C where we obtain 100% prediction accuracy compared with 78% accuracy. More detailed results and discussion can be found in (Islam et al., 2006a, 2006b, 2006c).

FUTURE TRENDS

In this work, we estimated the tumor-specific gene expression measure exploiting an ANOVA model with the parameters as fixed effects. As discussed in the future trends section of the chapter entitled: CNS Tumor Prediction Using Gene Expression Data Part I, we may consider all of the effects in our ANOVA model as random and specify the random effect parameters in a Bayesian framework. Once the distributions of tumor-specific gene expression measures are obtained with satisfactory confidence, more representative tumor prototypes may be obtained and a more accurate tumor prediction scheme can be formalized. Representing the tumor prototypes with a mixture of Gaussian models may provide a better representation. In that case, finding the number of components in the mixture is another research question left for future work.

CONCLUSION

For automatic tumor prediction, we have proposed a novel algorithm for building CNS tumor prototypes

---

Algorithm 1 (*DATA*) to build tumor prototype

DATA contains the expression levels of the marker genes for all the patient samples; where each row represents different gene expression values and each column represents different patient samples.

1.   Cluster genes into *K* partitions, *C= {C₁, C₂,…,Cₖ}*, using Complete Linkage Hierarchical approach and *correlation coefficient* as the pair wise similarity measure.

2.   For each cluster C_i
    a.   Compute the principal components (PCs).
    b.   If the cluster cardinality is greater than 3, project the genes onto the first 3 PCs else project the genes onto the first PC.
    Note: the projected vectors are considered as eigengene expressions.

3.   Considering the eigengene expressions as feature vectors, cluster each histological tumor group into subgroups exploiting SOM.

4.   The set of centroid of the corresponding SOM grid is designated as the tumor prototype of that particular histological tumor type.

C

Table 1. Confusion matrix for five categories of tumor samples

	Medulloblastoma	Glioma	AT/RTs	Normal	PNET	Recall
Medulloblastoma	9	0	1	0	0	0.9
Glioma	0	9	1	0	0	0.9
AT/RTs	0	0	10	0	0	1.0
Normal	0	0	1	3	0	0.75
PNET	0	1	0	0	7	0.88
Precision	1.0	0.9	0.77	1.0	1.0	

Overall Classification Accuracy: 90%

Table 2. Comparison table

Data Group	Number of Categories	Number of Samples	Classification Accuracy (our method)	Classification Accuracy (Pomeroy et al., 2002)
A	5	42	90%	83%
B	2	34	100%	97%
C	2	60	100%	78%

based on Affymetrix microarray gene expression values. We derived prototypes for different histological tumor types considering their genotype heterogeneity within groups. The eigengenes encode the correlations among gene expressions into the prototypes. Also, the eigengene expression measures are derived from estimated tumor-specific gene expression measures that are free from other unwanted sources of variations. We proposed a novel, seamless procedure that integrates normalization and tumor prediction considering both probable subgroupings within known tumor types and probable correlations among genes. The strong compliance of our results with the current molecular classification of the available tumor types suggests that our proposed model and its unique solution have significant practical value for automatic CNS tumor detection.

REFERENCES

Alizadeh, A.A., Eisen, M.B., & Davis, R.E., Ma, C., Lossos, I.S. Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson, J., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburge, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., & Staudt, L.M. (2000). Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling. Nature, 403, 503–511.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., & Levine, A. (1999). Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. Proceedings of National Academic of Science, 96(12), 6745–6750.

Dembélé, D., & Kastner, P. (2003). Fuzzy C-Means Method for Clustering Microarray Data. Bioinformatics, 19(8), 973–980.

Dettling, M., & Buhlmann, P. (2002). Supervised Clustering of Genes. Genome Biology, 3(12), 1-15.

Eisen, M., Spellman, P., Brown, P., & Botstein, D. (1998). Cluster Analysis and Display of Genome-Wide Expression Patterns. Proceedings of National Academic of Science, 95(14), 863–868.

Golub, T., Slonim, D ., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., & Lander, E. (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science, 286, 531–537.

Herwig, R., Poustka, A., Müller, C., Bull, C., Lehrach, H., & O'Brien, J. (1999). Large-Scale Clustering of cDNA-Fngerprinting Data. Genome Research, 9(11), 1093–1105.

Islam, A., & Iftekharuddin, K. (2005). Identification of Differentially Expressed Genes for CNS Tumor Groups and Their Utilization in Prognosis. Memphis Bio-Imaging Symposium.

Islam, A., Iftekharuddin, K.M., & Russomanno, D. (2006a). Visualizing the Differentially Expressed Genes. Proceedings of the Photonic Devices and Algorithms for Computing VIII, SPIE, 6310, 63100O.

Islam, A., Iftekharuddin, K.M., & Russomanno, D. (2006b). Marker Gene Selection Evaluation in Brain Tumor Patients Using Parallel Coordinates. Proceedings of the International Conference on Bioinformatics & Computational Biology, 172-174.

Islam, A., Iftekharuddin, K.M., & George, E.O. (2006c). Gene Expression Based CNS Tumor Prototype for Automatic Tumor Detection. Proceedings of the Fortieth Asilomar Conference on Signals, Systems and Computers, 846-850.

Kohonen, T. (1987). Self-Organization and Associative Memory, Springer-Verlag, 2nd Edition.

Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., Sturla, L.M., Angelo, M., McLaughlin, M.E., Kim, J.Y., Goumnerova, L.C., Black, P.M., Lau, C., Allen, J.C., Zagzag, D., Olson, J.M., Curran, T., Wetmore, C., Biegel, J.A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D.N., Mesirov, J.P., Lander, E.S., & Golub, T.R. (2002). Prediction of Central Nervous System Embryonal Tumor Outcome Based on Gene Expression. Nature, 415, 436-442.

Speed, T. (2003). Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall/CRC.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E., & Golub, T. (1999). Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation. Proceedings of National Academic of Science, 96(6), 2907–2912.

Taylor, M.D., Poppleton, H., Fuller, C., Su, X., Liu, Y., Jensen, P., Magdaleno, S., Dalton, J., Calabrese, C., Board, J., MacDonald, T., Rutka, J., Guha, A., Gajjar, A., Curran, T., & Gilbertson, R.J. (2005). Radial Glia Cells are Candidate Stem Cells of Ependymoma. Cancer Cell, 8(4), 323-335.

KEY TERMS

DNA Microarray: Also known as a DNA chip, it is a collection of microscopic DNA spots, commonly representing single genes, arrayed on a solid surface by covalent attachment to chemically suitable matrices.

False Discovery Rate (FDR): FDR controls the expected proportion of false positives instead of controlling the chance of any false positives. A FDR threshold is determined from the observed p-value distribution from multiple single hypothesis tests.

Histologic Examination: The examination of tissue specimens under a microscope.

Kruskal-Wallis Test: This test is a nonparametric mean test which can be applied if the number of sample group is more than two, unlike the Wilcoxon Rank Sum Test.

Parallel Coordinates: A multidimensional data visualization scheme that exploits 2D pattern recognition capabilities of humans. In this plot, the axes are equally spaced and are arranged parallel to one another rather than being arranged mutually perpendicular as in the Cartesian scenario.

q-Values: A means to measure the proportion of FDR when any particular test is called significant.

Self-Organizing Maps (SOMs): A method to learn to cluster input vectors according to how they are naturally grouped in the input space. In its simplest form, the map consists of a regular grid of units and the units learn to represent statistical data described by model vectors. Each map unit contains a vector used to represent the data. During the training process, the

C

model vectors are changed gradually and then the map forms an ordered non-linear regression of the model vectors into the data space.

Wilcoxon Rank Sum Test: A nonparametric alternative to the two sample t-test which is based on the order in which the observations from the two samples fall**.**

Combining Classifiers and Learning Mixture-of-Experts

Lei Xu
Chinese University of Hong Kong, Hong Kong & Peking University, China

Shun-ichi Amari
Brain Science Institute, Japan

INTRODUCTION

Expert combination is a classic strategy that has been widely used in various problem solving tasks. A team of individuals with diverse and complementary skills tackle a task jointly such that a performance better than any single individual can make is achieved via integrating the strengths of individuals. Started from the late 1980' in the handwritten character recognition literature, studies have been made on combining multiple classifiers. Also from the early 1990' in the fields of neural networks and machine learning, efforts have been made under the name of ensemble learning or mixture of experts on how to learn jointly a mixture of experts (parametric models) and a combining strategy for integrating them in an optimal sense.

The article aims at a general sketch of two streams of studies, not only with a re-elaboration of essential tasks, basic ingredients, and typical combining rules, but also with a general combination framework (especially one concise and more useful one-parameter modulated special case, called α-integration) suggested to unify a number of typical classifier combination rules and several mixture based learning models, as well as max rule and min rule used in the literature on fuzzy system.

BACKGROUND

Both streams of studies are featured by two periods of developments. The first period is roughly from the late 1980s to the early 1990s. In the handwritten character recognition literature, various classifiers have been developed from different methodologies and different features, which motivate studies on combining multiple classifiers for a better performance. A systematical effort on the early stage of studies was made in (Xu,

Krzyzak & Suen, 1992), with an attempt of setting up a general framework for classifier combination. As re-elaborated in Tab.1, not only two essential tasks were identified and a framework of three level combination was presented for the second task to cope with different types of classifier's output information, but also several rules have been investigated towards two of the three levels, especially with Bayes voting rule, product rule, and Dempster-Shafer rule proposed. Subsequently, the rest one (i.e., rank level) was soon studied in (Ho, Hull, & Srihari, 1994) via Borda count.

Interestingly and complementarily, almost in the same period the first task happens to be the focus of studies in the neural networks learning literature. Encountering the problems that there are different choices for the same type of neural net by varying its scale (e.g., the number of hidden units in a three layer net), different local optimal results on the same neural net due to different initializations, studies have been made on how to train an ensemble of diverse and complementary networks via cross-validation-partitioning, correlation reduction pruning, performance guided re-sampling, etc, such that the resulted combination produces a better generalization performance (Hansen & Salamon, 1990; Xu, Krzyzak, & Suen, 1991; Wolpert, 1992; Baxt, 1992, Breiman, 1992&94; Drucker, et al, 1994). In addition to classification, this stream also handles function regression via integrating individual estimators by a linear combination (Perrone & Cooper, 1993). Furthermore, this stream progresses to consider the performance of two tasks in Tab.1 jointly in help of the mixture-of-expert (ME) models (Jacobs, et al, 1991; Jordan & Jacobs, 1994; Xu & Jordan, 1993; Xu, Jordan & Hinton, 1994), which can learn either or both of the combining mechanism and individual experts in a maximum likelihood sense.

Two stream studies in the first period jointly set up a landscape of this emerging research area, together

Table 1. Essential tasks and their implementations

Two Tasks (a quotation from Xu, Krzyzak and Suen, 1992)

Task 1: "How many and what type of classifiers should be used ~~for a specific application~~?, and for each classifier what type of features should we use?, as well as other problems that are related to the construction of those individual and complementary classifier".

Task 2: "How to combine the results from different existing classifiers so that a better result can be obtained?"

Two Styles of Implementations

Two Stage Implementation	**Joint Implementation**	
•*Task 1* is completed in advance, with the resulted classifiers being diverse and complementary. • Perform *Task 2* in one of three levels (Xu, Krzyzak and Suen, 1992).	Two tasks made jointly or alternatively	
	under a same criterion	**others**
Level 1 each classifier outputs an abstract label. — Level 2 each classifier outputs a subset of ranked labels. — Level 3 each classifier outputs a vector of measurements with each on how likely a label is.	• Mixture of experts (ME) (Jacobs, et al, 1991; Jordan & Jacobs, 1994); •Alternative ME (Xu & Jordan, 1993; Xu, Jordan & Hinton, 1994); • EM-RBF (Xu,1998); •Three layer nets, etc.	Stacking , Boosting, ..., etc (Breiman, 1992&94; Wolpert, 1992)
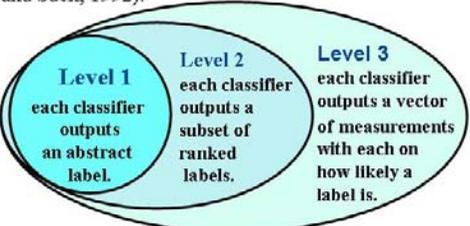
with a number of typical topics or directions. Thereafter, further studies have been further conducted on each of these typical directions. First, theoretical analyses have been made for deep insights and improved performances. For examples, convergence analysis on the EM algorithm for the mixture based learning are conducted in (Jordan & Xu, 1995; Xu & Jordan, 1996). In Tumer & Ghosh (1996), the additive errors of posteriori probabilities by classifiers or experts are considered, with variances and correlations of these errors investigated for improving the performance of a sum based combination. In Kittler, et al (1998), the effect of these errors on the sensitivity of sum rule vs product rule are further investigated, with a conclusion that summation is much preferred. Also, a theoretical framework is suggested for taking several combining rules as special cases (Kittler, 1998) , being unaware of that this framework is actually the mixture-of-experts model that was proposed firstly for combining multiple function regressions in (Jacobs, et al, 1991) and then for combining multiple classifiers in (Xu & Jordan, 1993). In addition, another theoretical study is made on six classifier fusion strategies in (Kuncheva, 2002). Second, there are further studies on Dempster-Shafer rule (Al-Ania, 2002) and other combing methods such as rank based, boosting based, as well as local accuracy estimates (Woods, Kegelmeyer, & Bowyer, 1997). Third, there are a large number of applications. Due to space limit, details are referred to Ranawana & Palade (2006) and Sharkey & Sharkey (1999).

A GENERAL ARCHITECTURE, TWO TASKS, AND THREE INGREDIENTS

We consider a general architecture shown in Fig.1. There are $\{e_j(x)\}_{j=1}^{k}$ experts with each $e_j(x)$ as either a classifier or an estimator. As shown in Tab.2, a classifier outputs one of three types of information, on which we have three levels of combination. The first two can be regarded as special cases of the third one that outputs a vector of measurements. A typical example is $[p_j(1|x),\ldots,p_j(m|x)]^T$ with each $1 \geq p_j(\ell|x) \geq 0$ expressing a posteriori probability that x is classified to the ℓ-th class. Also, $p_j(\ell|x) = p_j(y = \ell|x)$ can be further extended to $p_j(y|x)$ that describes a distribution for a regression $x \rightarrow y \in R^m$. In Figure 1, there is also a gating net that generates signals $\{\alpha_j(x)\}_{j=1}^{k}$ to modulate experts by a combining mechanism $M(x)$.

Figure 1. A general architecture for expert combination
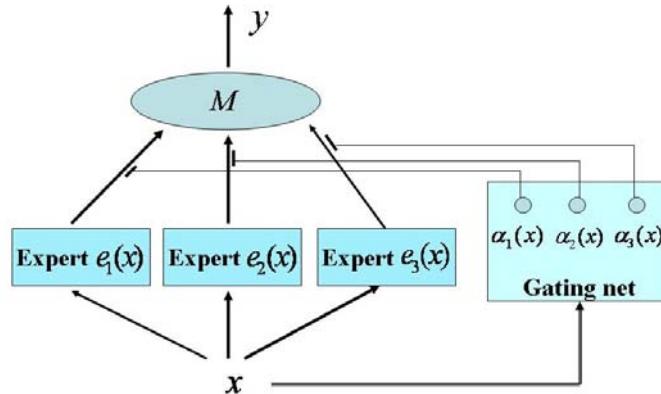
Table 2. Three levels of combination

Three Rules for Combination on Level 3		
Sum rule (Bayes Voting)	**Product rule**	**Dempster-Shafer rule**
Given k classifiers, the j-th classifier classifies x to y with a probability $p_j(y\mid x)=p(x\in C_y\mid e_j(x))$, we sum up to get a combination $p(y\mid x)=\frac{1}{k}\sum_{j=1}^{k}p_j(y\mid x)$ See eqn.(4) in (Xu, Krzyzak and Suen, 1992)	If k classifiers are independent, another combination is given by $p(x\in C_y)=\dfrac{\prod_{j=1}^{k}p(x\in C_y\mid e_j(x))}{\prod_{j=1}^{k}p(x\in C_y)}$ or concisely $p(y\mid x)=p^{1-k}(y)\prod_{j=1}^{k}p_j(y\mid x)$ See eqn.(31) in (Xu, Krzyzak and Suen, 1992)	$bel(A)=\sum_{B\subseteq A}m(B)$ $m(A)=\dfrac{\sum_{X\cap Y=A,\,A\neq\phi}m_\ell(X)m_{\ell'}(Y)}{\sum_{X\cap Y=\phi}m_\ell(X)m_{\ell'}(Y)}$ A_t denotes $x\in C_t$, $\theta=\{A_1,\ldots,A_m\}$ if $e_j(x)=\ell$, $m_j(A_t)=\varepsilon_r^{(j)}$ $\neg A_t=\theta-\{A_t\}$ $m_j(\neg A_t)=\varepsilon_s^{(j)}$ $m_j(\theta)=1-\varepsilon_r^{(j)}-\varepsilon_s^{(j)}$ See Sec.VI in (Xu, Krzyzak and Suen, 1992)

Based on this architecture, two essential tasks of expert combination could be still quoted from (Xu, Krzyzak & Suen, 1992) with a slight modification on Task 1, as shown in Tab.1, that the phrase `for a specific application?' should be deleted in consideration of the previously introduced studies (Hansen & Salamon, 1990; Xu, Krzyzak, & Suen, 1991; Wolpert, 1992; Baxt, 1992, Breiman, 1992&94; Drucker, et al, 1994; Tumer & Ghosh, 1996).

Insights can be obtained by considering three basic ingredients of two streams of studies, as shown in Fig.2. Combinatorial choices of different ingredients lead to different specific models for expert combination, and differences in the roles by each ingredient highlight the different focuses of two streams. In the stream

of neural networks and machine learning, provided with a structure for each $e_j(x)$, a gating structure, and a combining structure $M(x)$, all the rest unknowns are determined under guidance of a learning theory in term of minimizing an error cost. Such a minimization is implemented via an optimizing procedure by a learning algorithm, based on a training set $\{x_t,y_t\}_{t=1}^{N}$ that teaches a target y_t for each mapping $x_t\to R^m$. While in the stream of combing classifiers, all $\{p_j(y\mid x)\}_{j=1}^{k}$ are known without unknowns left to be specified. Also, M is designed according to certain heuristics or principles, with or without help of a training set, and studies are mainly placed on developing and analyzing different combining mechanisms, for which we will further dis-

cuss subsequently. The final combining performance is empirically evaluated by the misclassification rate, but there is no effort yet on developing a theory for one *M* that minimizes the misclassification rate or a cost function, though there are some investigations on how estimated posteriori probabilities can be improved by a sum rule and on error sensitivity of estimated posteriori probabilities (Tumer & Ghosh, 1996; Kittler, et al, 1998). This under-explored direction also motivate future studies subsequently.

f-COMBINATION

The arithmetic, geometric, and harmonic mean of non-negative number $b_j \geq 0, j = 1, \ldots, k$ has been further extended into one called:

f-mean $m_f = f^{-1}(\sum_{j=1}^{k} \alpha_j f(b_j))$,

where *f*(*r*) is a monotonic scalar function, and

$\alpha_j > 0 \sum_{j=1}^{k} \alpha_j = 1$

(Hardy, Littlewood, & Polya, 1952).

We can further generalize this *f-mean* to the general architecture shown in Fig.1, resulting in the following *f*-combination:

$$M(x) = f^{-1}(\sum_{j=1}^{k} \alpha_j(x)f(p_j(y \mid x))), \text{ or}$$

$$f(M(x)) = \sum_{j=1}^{k} \alpha_j(x)f(p_j(y \mid x))$$

where

$\alpha_j > 0 \sum_{j=1}^{k} \alpha_j(x) = 1$.

In the following, we discuss to use it as a general framework to unify not only typical classifier combining rules but also mixture-of-expert learning and RBF net learning, as shown in Tab.3.

We observe the three columns for three special cases of *f*(*r*). The first column is the case *f*(*r*) = *r*, we return to the ME model:

$$M(x) = \sum_{j=1}^{k} \alpha_j(x)p_j(y \mid x),$$

which was proposed firstly for combining multiple regressions in (Jacobs, et al, 1991) and then for combining classifiers in (Xu & Jordan, 1993). For different special cases of $\alpha_j(x)$, we are lead to a number of existing typical examples. As already pointed out in (Kittler, et al, 1998), the first three rows are four typical classifier combining rules (the 2nd row directly applies to the min-rule too). The next three rows are three types of ME learning models, and a rather systematic summary

Figure 2. Three basic ingredients
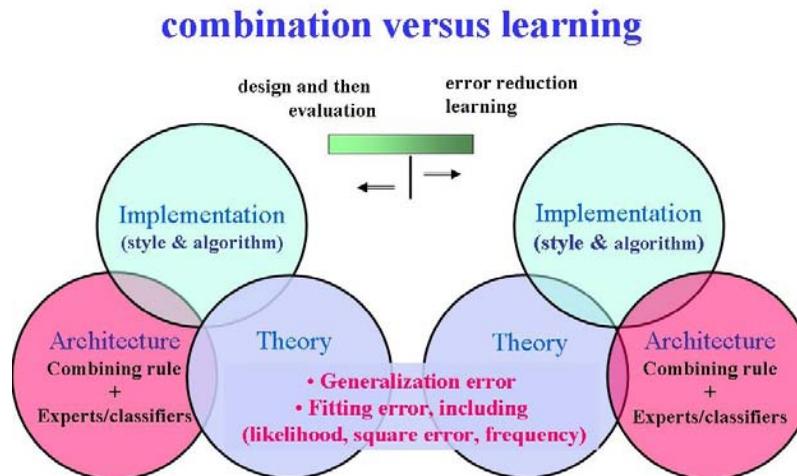
Table 3. Typical examples (including several existing rules and potential topics)
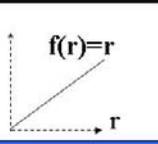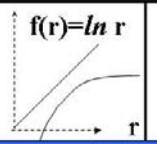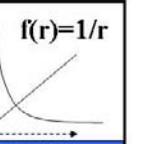
$f(M)$ \ $f(r)$ \ $\alpha_i(x)$	$f(r)=r$	$f(r)=ln\ r$	$f(r)=1/r$
$\alpha_j(x)=0$, except $\alpha_{j^*}(x)=1$ $\ j^* = \begin{cases}\arg\max_j p_j(y\mid x),\ (a),\\ \arg\min_j p_j(y\mid x),\ (b).\end{cases}$	(a) max-rule (b) min-rule	(a) max-rule (b) min-rule	(a) max-rule (b) min-rule
$\alpha_j(x)=1/k$	Average Bayes or Bayes voting (Xu, Krzyzak & Suen, 1992)	Product rule (Xu, Krzyzak & Suen, 1992; Kittler, et al, 1998; Hinton, 2002)	Harmonic mean
$\alpha_j(x)=\alpha_j=\dfrac{\sigma_j^{-2}}{\sum_{j=1}^{k}\sigma_j^{-2}}.$	Mixture using variances (MUV) (Perrone&Cooper, 1993)	To be explored	To be explored
$\alpha_j(x)=\dfrac{e^{g_j(x,\phi)}}{\sum_{\ell=1}^{k}e^{g_\ell(x,\phi)}}$	Mixture-of-experts (ME) (Jacobs, et al, 1991)	To be explored	To be explored
$\alpha_j(x)=\dfrac{\beta_j p(x\mid\phi_j)}{\sum_{\ell=1}^{k}\beta_\ell p(x\mid\phi_\ell)}$	Alternative ME (Xu, Jordan& Hinton, 1994)	To be explored	To be explored
$\alpha_j(x)=\dfrac{\beta_j G(x\mid\mu_j,\Sigma_j)}{\sum_{\ell=1}^{k}\beta_\ell G(x\mid\mu_\ell,\Sigma_\ell)},$ $subject\ to\ \beta_j/\sqrt{\mid\Sigma_j\mid}=const$	Extended Normalized RBF (Xu, 1998)	To be explored	To be explored
$\alpha_j(x)=\sigma_j^{-2}(x)\big/\sum_{j=1}^{k}\sigma_j^{-2}(x)$	Belief net based MUV (Lee, et al, 2006)	To be explored	To be explored

is referred to Sec. 4.3 in (Xu, 2001). The last row is a recent development of the 3rd row.

The 2nd row of the 2nd column is the geometric mean:

$$M(x) = \sqrt[k]{\prod_{j=1}^{k} p_j(y\mid x)},$$

which is equal to the product rule (Xu, Krzyzak and Suen, 1992; Kittler, et al, 1998, Hinton, 2002) if each a priori is equal, i.e., $\alpha_j(x) = 1/m$. Generally if $\alpha_j(x) \neq 1/m$, there is a difference by a scaling factor $\alpha_j(x)^{1/k-1}$. The product rule works in a probability theory sense under a condition that classifiers are mutually independent. In (Kittler, et al, 1998), attempting to discuss a number of rules under a unified system, the sum rule is approximately derived from the product rule, under an extra condition that is usually difficult to satisfy. Actually, such an imposed link between the product rule and the sum rule is unnecessary, the sum:

$$M(x) = \sum_{j=1}^{k}\alpha_j(x)p_j(y\mid x)$$

is just a marginal probability

$$\sum_{j=1}^{k} p(y,j\mid x),$$

which is already in the framework of probability theory. That is, both the sum rule and the product rule already coexist in the framework of probability theory.

On the other hand, it can be observed that the sum:

$$\sum_{j=1}^{k}\alpha_j(x)\ln p_j(y\mid x)$$

is dominated by a $p_j(y\mid x)$ if it is close to 0. That is, this combination expects that every expert should cast enough votes, otherwise the combined votes will be still very low just because there is only one that casts a very low vote. In other words, this combination can

C

be regarded as a relaxed logical AND that is beyond the framework of probability theory when $\alpha_j(x) \neq 1/m$. However, staying within the framework of probability theory does not mean that it is better, not only because it requires that classifiers are mutually independent, but also because there lacks theoretical analysis on both rules in a sense of classification errors, for which further investigations are needed.

In Tab.2, the 2nd row of the third column is the harmonic mean. It can be observed that the problem of combining the degrees of support is changed into a problem of combining the degrees of disagree. This is interesting. Unfortunately, efforts of this kind are seldom found yet. Exceptionally, there are also examples that can not be included in the *f*-combination, such as Dempster-Shafer rule (Xu, Krzyzak and Suen, 1992; Al-Ania, 2002) and rank based rule (Ho, Hull, Srihari, 1994).

α-INTEGRATION

After completed the above *f*-combination, the first author becomes aware of the work by (Hardy, Littlewood, & Polya, 1952) through one coming paper (Amari, 2007) that studies a much concise and more useful one-parameter modulated special case called α-integration. With help of a concrete mathematical foundation from an information geometry perspective. Imposing an additional but reasonable nature that the *f-mean* should be linear scale-free, i.e.:

$$cm_f = f^{-1}(\sum_{j=1}^{k}\alpha_j f(cb_j))$$

for any scale *c,* alternative choices of *f(r)* reduces into the following only one:

$$f_\alpha(r) = \begin{cases} r^{0.5(1-\alpha)}, & \alpha \neq 1, \\ \ln r, & \alpha = 1. \end{cases}$$

It is not difficult to check that

$$f_\alpha(r) = \begin{cases} r, & \alpha = -1, \\ \ln r, & \alpha = 1, \\ 1/r, & \alpha = 3. \end{cases}$$

Thus, the discussions on the examples in Tab.2 are applicable to this $f_\alpha(r)$. Moreover, the first row in Tab.2 holds when $\alpha = -\infty$ and $\alpha = +\infty$ for whatever a gating net, which thus includes two typical operators of the fuzzy system as special case too. Also, the family is systematically modulated by a parameter $-\infty \leq \alpha \leq +\infty$, which provides not only a spectrum from the most optimistic integration to the most pessimistic integration as varying from $\alpha = -\infty$ to $\alpha = +\infty$ but also a possibility of adapting α for a best combining performance.

Furthermore, Amari (2007) also provides a theoretical justification that α-integration is optimal in a sense of minimizing a weighted average of α-divergence. Moreover, it provides a potential road for studies on combining classifiers and learning mixture models from the perspective of information geometry.

FUTURE TRENDS

Further studies are expected along several directions as follows:

- Empirical and analytical comparisons on performance are needed for those unexplored or less explored items in Tab.2.
- Is there a best structure for $\alpha_j(x)$? comparisons need to be made on different types of $\alpha_j(x)$, especially the ones by the MUV type in the last row and the ME types from the 4th to the 7th rows,
- Is it necessary to relax the constraint:

$$\alpha_j(x) > 0, \ \sum_{j=1}^{k}\alpha_j(x) = 1,$$

  e.g., removing non-negative requirement and to relax the distribution $p_j(y\,|\,x)$ to other types of functions ?
- How weights $\alpha_j(x)$ can be learned under a generalization error bound.
- As discussed in Fig.2, classifier combination and mixture based learning are two aspects with different features. How to let each part to take their best roles in an integrated system?

CONCLUSION

Updating the purpose of (Xu, Krzyzak & Suen, 1992), the article provides not only a general sketch of studies

on combining classifiers and learning mixture models, but also a general combination framework to unify a number of classifier combination rules and mixture based learning models, as well as a number of directions for further investigations.

ACKNOWLEDGMENT

REFERENCES

Al-Ania , A., (2002), A New Technique for Combining Multiple Classifiers using The Dempster-Shafer Theory of Evidence, Journal of Artificial Intelligence Research 17, 333-361.

Amari, S., (2007), Integration of stochastic models by minimizing α-divergence, Neural Computation 19(10), 2780-2796.

Baxt, W. G. (1992), Improving the accuracy of an artificial neural network using multiple differently trained networks, Neural Computation 4, 772-780.

Breiman, L., (1992), Stacked Regression, Department of Statistics, Berkeley, TR-367.

Breiman, L., (1994), Bagging Predictors, Department of Statistics, Berkeley, TR-421.

Drucker,H., Cortes,C., Jackel,L.D., LeCun,Y.& Vapnik, V., (1994) Boosting and other ensemble methods, Neural Computation 6, 1289-1301.

Hardy, G.H., Littlewood, J.E., and Polya, G. (1952), Inequalities, 2nd edition, Cambridge University Press.

Hansen, L.,K. & Salamon, P. (1990), Neural network ensembles. IEEE Transactions Pattern Analysis Machine Intelligence 12(10), 993-1001.

Hinton, G.E.(2002), Training products of experts by minimizing contrastive divergence, Neural Computation 14, 1771-1800.

Ho, T.K., Hull ., J.J., Srihari, S., N., (1994), Decision combination in multiple classifier systems, IEEE Transactions Pattern Analysis Machine Intelligence 16(1), 66-75

Jacobs, R. A., Jordan, M. I. , Nowlan, S. J., & Hinton, G. E., (1991), Adaptive mixtures of local experts, Neural Computation 3, 79-87.

Jordan, M. I. , & Jacobs, R. A., (1994), Hierarchical mixtures of experts and the EM algorithm, Neural Computation 6, 181–214.

Jordan, M.I., & Xu, L., (1995), Convergence Results for The EM Approach to Mixtures of Experts Architectures, Neural Networks 8, 1409-1431.

Kittler, J., Hatef, M., Duinand, R., P., W. , Matas, J., (1998) On combining classifiers, IEEE Trans. Pattern Analysis Machine Intelligence 20(3), 226–239.

Kittler, J., (1998) , Combining classifiers: A theoretical framework, Pattern Analysis and Applications 1, 18-27.

Kuncheva, L.,I., (2002), A theoretical study on six classifier fusion strategies, IEEE Transactions Pattern Analysis Machine Intelligence 24(2), 281–286.

Lee, C., Greiner, R., Wang, S., (2006), Using query-specific variance estimates to combine Bayesian classifiers, Proc. of the 23rd International Conference on Machine Learning (ICML), Pittsburgh, June (529-536).

Perrone, M.P. , & Cooper, L.N.,(1993), When networks disagree: Ensemble methods for neural networks, Neural Networks for Speech and Image Processing, Mammone, R., J., editor, Chapman Hall.

Ranawana, R. & Palade, V., (2006), Multi-Classifier Systems: Review and a roadmap for developers, International Journal of Hybrid Intelligent Systems 3(1), 35 – 61.

Shafer, G., (1976), A mathematical theory of evidence, Princeton University Press.

Sharkey, A.J.& Sharkey, N.E., (1997), Combining diverse neural nets, Knowledge Engineering Review 12(3), 231–247.

Sharkey, A.J.& Sharkey, N.E., (1999.), Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems, Springer-Verlag, NewYork, Inc.

Tumer, K. & Ghosh, J. (1996), Error correlation and error reduction in ensemble classifiers, Connection Science, 8 (3/4), 385-404.

Wolpert, D.,H., (1992), Stacked generalization, Neural Networks 5(2), 241-260.

Woods,K., Kegelmeyer,K.P., & Bowyer, K., (1997), Combination of multiple classifiers using local accuracy estimates, IEEE Transactions Pattern Analysis Machine Intelligen 19(4), 405-410.

Xu, L (2007), A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving, Pattern Recognition 40, 2129–2153.

Xu, L (2001), Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM Models, International Journal of Neural Systems 11(1), 43-69.

Xu, L., (1998), RBF Nets, Mixture Experts, and Bayesian Ying-Yang Learning, Neurocomputing 19, 223-257.

Xu, L. & Jordan, M.I., (1996), On convergence properties of the EM algorithm for Gaussian mixtures, Neural Computation, 8(1), 129-151.

Xu, L., Jordan, M.I., & Hinton, G.E., (1995), An Alternative Model for Mixtures of Experts, *Advances in Neural Information Processing Systems 7*, Cowan, Tesauro, and Alspector, editors, MIT Press, 633-640

Xu, L., Jordan, M.I., & Hinton, G.E., (1994), A Modified Gating Network for the Mixtures of Experts Architecture, *Proc. WCNN94*, San Diego, CA, (2) 405-410.

Xu, L. & Jordan, M.I., (1993), EM Learning on A Generalized Finite Mixture Model for Combining Multiple Classifiers, Proc. of WCNN93, (IV) 227-230.

Xu, L., Krzyzak, A. & Sun, C.Y., (1992), Several methods for combining multiple classifiers and their applications in handwritten character recognition, IEEE Transactions on System, Man and Cybernetics 22, 418-435.

Xu, L., Krzyzak, A. & Sun, C.Y., (1991), Associative Switch for Combining Multiple Classifiers, Proc. of IJCNN91, July 8-12. Seattle, WA, (I) 43-48.

Xu, L. Oja, E., & Kultanen, P., (1990), A New Curve Detection Method Randomized Hough transform (RHT), Pattern Recognition Letters 11, 331-338.

KEY TERMS

Conditional Distribution $p(y/x)$: Describes the uncertainty that an input x is mapped into an output y that simply takes one of several labels. In this case, x is classified into the class label y with a probability $p(y|x)$. Also, y can be a real-valued vector, for which x is mapped into y according density distribution $p(y|x)$.

Classifier Combination: Given a number of classifiers, each classifies a same input x into a class label, and the labels maybe different for different classifiers. We seek a rule $M(x)$ that combines these classifiers as a new one that performs better than anyone of them.

Sum Rule (Bayes Voting): A classifier classifies x to a label y can be regarded as casting one vote to this label, a simplest combination is to count the votes received by every candidate label. The j-th classifier classifies x to a label y with a probability $p_j(y|x)$ means that one vote is divided to different candidates in fractions. We can sum up:

$$\sum_j p_j(y|x)$$

to count the votes on a candidate label y, which is called Bayes voting since $p(y|x)$ is usually called Bayes posteriori probability.

Product Rule: When k classifiers $\{e_j(x)\}_{j=1}^k$ are mutually independent, a combination is given by

$$p(x \in C_y | x) = p(x \in C_y) \frac{\prod_{j=1}^{k} p(x \in C_y | e_j(x))}{\prod_{j=1}^{k} p(x \in C_y)}$$

or concisely

$$p(y \mid x) = p^{1-k}(y) \prod_{j=1}^{k} p_j(y \mid x),$$

which is also called product rule.

Mixture of Experts: Each expert is described by a conditional distribution $p_j(y \mid x)$ either with y taking one of several labels for a classification problem or with y being a real-valued vector for a regression problem. A combination of experts is given by:

$$M(x) = \sum_{j=1}^{k} \alpha_j(x) p_j(y \mid x), \ \alpha_j(x) = p(j \mid x) > 0, \ \sum_{j=1}^{k} \alpha_j(x) = 1,$$

which is called a mixture-of-experts model. Particularly, for y in a real-valued vector, its regression form is

$$E(y \mid x) = \sum_{j=1}^{k} \alpha_j(x) f_j(x), \ f_j(y) = \int y p_j(y \mid x) dy.$$

f-Mean: Given a set of non-negative numbers $b_j \geq 0, j = 1, \dots, k$, the f-mean is given by:

$$m_f = f^{-1}(\sum_{j=1}^{k} \alpha_j f(b_j)),$$

where $f(r)$ is a monotonic scalar function and

$$\alpha_j > 0 \sum_{j=1}^{k} \alpha_j = 1.$$

Particularly, one most interesting special case is that $f(r)$ satisfies

$$cm_f = f^{-1}(\sum_{j=1}^{k} \alpha_j f(cb_j))$$

for any scale c, which is called f_a-mean.

Performance Evaluation Approach: It usually works in the literature on classifier. Combination, with a chart flow that considering a set of classifiers $\{e_j(x)\}_{j=1}^{k}$ → designing a combining mechanism $M(x)$ according to certain principles → evaluating performances of combination empirically via misclassification rates, in help of samples with known correct labels.

Error-Reduction Approach: It usually works in the literature on mixture based learning, where what needs to be pre-designed is the structures of classifiers or experts, as well as the combining structure $M(x)$ with unknown parameters. A cost or error measure is evaluated via a set of training samples, and then minimized through learning all the unknown parameters.

Commonsense Knowledge Representation I

Phillip Ein-Dor
Tel-Aviv University, Israel

INTRODUCTION

Significant advances in artificial intelligence, including machines that play master level chess, or make medical diagnoses, highlight an intriguing paradox. While systems can compete with highly qualified experts in many fields, there has been much less progress in constructing machines that exhibit simple commonsense, the kind expected of any normally intelligent child. As a result, commonsense has been identified as one of the most difficult and important problems in AI (Doyle, 1984; Waltz, 1982).

BACKGROUND

The Importance of Commonsense[1]

It may be useful to begin by listing a number of reasons why Commonsense is so important:

1. Any general natural language processor must possess the commonsense that is assumed in the text.
2. In building computerized systems, many assumptions are made about the way in which they will be used and the users' background knowledge. The more commonsense that can explicitly be built into systems, the less will depend on the implicit concurrence of the designer's commonsense with that of the user.
3. Many expert systems have some commonsense knowledge built into them, much of it reformulated time and again for similar systems. It would be advantageous if commonsense knowledge could be standardized for use in different systems.
4. Commonsense has a large element that is environment and culture specific. A study and formalization of commonsense knowledge may permit people of different cultures to better understand one another's assumptions.

Defining Commonsense

No attempt will be made here to define commonsense rigorously. Intuitively, however, commonsense is generally meant to include the following capabilities, as defined for any given culture:

a. knowing the generally known facts about the world,
b. knowing, and being able to perform, generally performed behaviors, and to predict their outcomes,
c. being able to interpret or identify commonly occurring situations in terms of the generally known facts – i.e. to understand what happens,
d. the ability to relate causes and effects,
e. the ability to recognize inconsistencies in descriptions of common situations and behaviors and between behaviors and their situational contexts,
f. the ability to solve everyday problems.

In summary, commonsense is the knowledge that any participant in a culture expects any other participant in that culture to possess, as distinct from specialized knowledge that is possessed only by specialists.

The necessary conditions for a formalization to lay claim to representing commonsense are implicit in the above definition; a formalism must exhibit at least one of the attributes listed there. Virtually all work in the field has attempted to satisfy only some subset of the commonsense criteria.

COMMONSENSE REPRESENTATION FORMALISMS

In AI research, work on common sense is generally subsumed under the heading of Knowledge Representation. The objective of this article is to survey the various formalisms that have been suggested for representing commonsense knowledge.

Four major knowledge representation schemes are discussed in the literature - production rules, semantic nets, frames, and logic. Production systems are frequently adopted in building expert systems. Virtually all the discussions of commonsense representations, however, are in terms of semantic net, frame-like, or logic systems. These schemes are applied within three main paradigms for commonsense representation—propositional, truth maintenance, and dispositional (see Figure 1). Very briefly, propositional models are descriptions of representations of things or concrete facts. When the knowledge represented is imprecise or variable, propositional formalisms are no longer sufficient and one needs to consider the beliefs about the world engendered by the system's current state of knowledge, and to allow for changes in those beliefs as circumstances dictate; this is the nature of belief or truth maintenance systems. Finally, when the knowledge is both imprecise and not factual, but relates rather to feelings, insights and understandings, the dispositional representations are evoked.

Within each representational paradigm, there are a number of specific formalisms. Figure 1 indicates the existence of eight different knowledge representation formalisms. Each of these formalisms is presented via discussion of one or more representatives.

The need for different types of formalisms, the difficulty in representing multiple domain knowledge, psychological theories of various levels of conscious-

ness, the physiological evidence of different levels of the brain and their association with specific functions, the functional specialization of specific areas of the brain, and similar evidence concerning the two sides of the brain all support the view of the mind, or self, as composed of a considerable number of cooperating subagents, to which Minsky (1981) refers as a society of mind. It is useful to keep this concept in mind while studying the variety of representation schemes; it suggests that a number of such formalisms may coexist in any rational agent and little can be gained by attempts to choose the "right" formalism in any general sense.

PROPOSITIONAL MODELS

Virtually all the propositional models of commonsense knowledge are perceived as consisting of nodes that are associated with words or tokens representing concepts. The nodes are hierarchically structured, with lower level nodes elaborating or representing instantiations of higher-level nodes; the higher-level nodes impart their properties to those below them, which are said to inherit those properties. Thus, all the propositional models are hierarchically structured networks consisting of nodes and arcs joining the nodes. From this point, the representational structures begin to diverge according to the distribution of information between the arcs and the nodes. At one extreme, nodes are self-contained

Figure 1. Commonsense knowledge representation formalisms
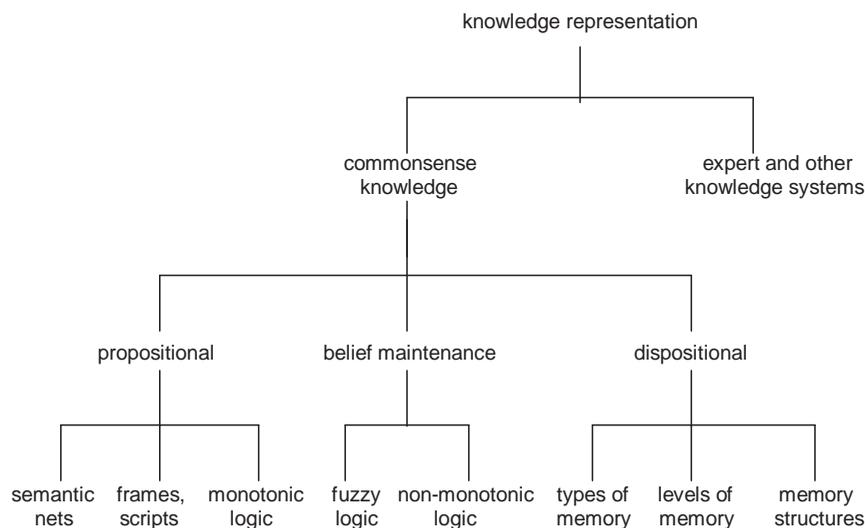
descriptions of concepts with only hierarchical relations between concepts expressed in the arcs; frames and scripts are of this nature. At the other extreme, nodes contain only names and the descriptive content resides in a multiplicity of types of relations, which give meaning to the names. This last form is generally referred to as a semantic net. Between the extremes lie representations in which the descriptive knowledge is distributed between nodes and relations.

The representations described may be discussed as theoretical models or as computer implementations, which generally derive from one of the theoretical models (see Commonsense Knowledge Representation II - Implementation).

Semantic Nets

Semantic nets were developed to represent propositions and their syntactic structure rather than the knowledge contained in the propositions. A semantic net consists of triples of nodes, arcs joining nodes, and labels (Rich, 1983). The words of a proposition are contained in the nodes while its syntactic structure is captured by the labeled arcs.

Frames

Within the community utilizing frames (or schema), it seems to be universally agreed that frames define concepts via the contents of slots, which specify components of the concepts. The content of a slot may be an instantiation, a default or usual value, a condition, or a sub-schema. The last of these slot contents generates the generally accepted hierarchical nature of concept representations.

Scripts

In his seminal paper on frames, Minsky (1975) suggests that frames are not exactly the same as Schank's (1981) scripts. The major difference is in the concepts represented. Scripts describe temporal sequences of events represented by the sequence of slots in a script. Here the slot structure is significant. For frames describing objects the exact order of slots is probably not significant. The temporal ordering of slots is necessary in many other representations, including those for most behaviors.

In summary, frames, scripts, and semantic nets are logically equivalent. All three are hierarchies consisting of nodes and relations; they are differentiated by the location of information in the network. Scripts are further differentiated by the temporal and causal ordering of slots within nodes.

Predicate Calculus

The first order predicate calculus is a method for representing and processing propositional knowledge and was designed expressly for that purpose. There are many workers in the field who view this formalism as the most appropriate for commonsense knowledge representation.

In discussing logic as a vehicle for knowledge representation, one should distinguish between the use of logic as the representational formalism, and the use of logistic languages for implementation. Logistic languages can implement any representational formalism, logistic or other. Another distinction that needs to be made is between the use of logic for inference and its use for representation (Hayes, 1979). Thus, one might apply logical inference to any knowledge representation, provided that it is amenable to that kind of manipulation; knowledge expressed in predicate calculus is the most amenable to such manipulation.

A vigorous debates in AI centered on the choice of frames or logic for commonsense knowledge representation. In favor of frames were those who attempted to implement knowledge representation schemes (cf. Hayes, 1979). The argument is twofold: first, logistic formalisms do not have sufficient expressive power to adequately represent commonsense knowledge. This applies particularly to the dynamic adjustment of what is held to be true as new information is acquired. Classical logic does not allow for reinterpretation as new facts are learned - it is monotonic. Secondly, this argument posits that human knowledge representation does not follow the rules of formal logic, so that logic is psychologically inadequate in addition to being expressively inadequate.

The logicians reply to the first argument is that anything can be expressed in logic, but this may sometimes be difficult; this school does not agree to the claim of greater expressive power for the frame paradigm. Hayes (1979) claimed that "most of 'frames' is just a new syntax for parts of first-order logic". Thus, all the representational formalisms seem to have much in

common, even if they are not completely equivalent. The reply to the second argument is that the object of study is Artificial Intelligence so that the formalisms adopted need not resemble human mechanisms.

One point of agreement that seems to have emerged between the two sides is that the classical predicate calculus indeed lacks sufficient expressive power. Evidence of this is a number of attempts to expand the logical formalisms in order to provide sufficient expressiveness, especially by alleviating some of the restrictions of monotonicity.

The essence of monotonicity is that if a theory A is expanded by additional axioms to become B, then all the theorems of A are still theorems of B. Thus, conclusions in monotonic logic are irreversible and this constraint must be relaxed if a logical inference system is to be able to reevaluate its conclusions on learning new facts. The conclusions of a non-monotonic logic are regarded as beliefs or dispositions based on what is currently known, and amenable to change if necessary.

The attempts to overcome the limitations of monotonicity include non-monotonic reasoning McCarthy (1980), non-monotonic logics (McDermott and Doyle, 1980; Reiter 1980), fuzzy logic (Zadeh, 1983), and a functional approach (Levesque, 1984) represented by the KL-One (Woods, 1983) and KRYPTON languages (Brachman et al., 1983). It should be noted that non-monotonic logic proceeds not by changing the logic representation, but rather by strengthening the reasoning by which representations are evaluated.

LOGISTIC REPRESENTATIONS: BELIEF MAINTENANCE SYSTEMS

Fuzzy Logic (Zadeh, 1983)

The extensive work by Zadeh in applying fuzzy logic to knowledge representation is based on the premise that predicate calculus is inadequate for commonsense knowledge representation because it does not allow for fuzzy predicates (e.g. small, cheap) or fuzzy quantifiers (e.g. many, most) as in the phrase "most small birds fly".

Fuzzy logic proceeds in the representation of dispositions by the following steps:

1. A proposition is regarded as a collection of (usually implicit) fuzzy constraints.
2. An explanatory database contains lists of samples of the subject of the proposition together with the degree to which each predicate or constraint is fulfilled. These data are used to compute test scores of the degree to which each constraint is met. These test-scores then become the meaning of the fuzzy constraints.

Circumscription (McCarthy, 1980)

Circumscription is a form of non-monotonic reasoning (as distinct from non-monotonic logic) which reduces the context of a sentence to anything that is deducible from the sentence itself, and no more. Without such a mechanism, any statement may invoke all the knowledge the system possesses that is associated with whatever the topic of the statement is, much of which would probably be irrelevant. The question

"Crows and canaries are both birds; why is Tweetie afraid of crows?"

could give rise to consideration of the facts that ostriches and penguins are also birds and that ostriches don't fly and put their heads in the sand while penguins swim and eat fish, all of which is irrelevant to the problem in hand. The purpose of circumscription is to limit evaluation of problem statements to the facts of the statement - what McCarthy describes as the ability to jump to conclusions.

Default Reasoning (Reiter, 1980)

Reiter's Default Reasoning is based on the first order predicate calculus and attempts to solve a particular problem that arises in expressing commonsense knowledge in the classical formalism. The problem is that of drawing conclusions on the basis of incomplete knowledge when the missing information can be assumed, provided there is no evidence to the contrary. Such assumptions are possible because much real world knowledge about classes of objects is almost always true. Thus, most birds are capable of flight with a few exceptions such as penguins, ostriches, and kiwis, and provided they do not suffer from specific inabilities such as death or feet embedded in concrete.

Non-Monotonic Logic (McDermott & Doyle, 1980)

Non-monotonic logic attempts to formalize the representation of knowledge in such a way that the models maintained on the basis of the premises supplied can change, if necessary, as new premises are added. Thus, rather than determining the truth or falsity of statements as in classical logic, non-monotonic logics hold beliefs based on their current level of knowledge.

DISPOSITIONAL MODELS

Dispositional models attempt to establish the relationship between knowledge representation and memory. To paraphrase Schank (1981), understanding sentences involves adding information, generally commonsense, not explicit in the original sentence. Thus, a memory full of facts about the commonsense world is necessary in order to understand language. Furthermore, a model, or set of beliefs is necessary to provide expectations or explanations of an actor's behavior. Adding beliefs to a representation changes the idea of inference from simply adding information to permit parsing sentences to integrating data with a given memory model; this leads to the study of facts and beliefs in memory.

K-Lines (Minsky, 1981)

Minsky's major thesis is that the function of memory is to recreate a state of mind. Each memory must embody information that can later serve to reassemble the mechanisms that were active when the memory was formed.

Memory is posited to consist, at the highest level, of a number of loosely interconnected specialists - a "society of mind". Each of the specialists, in turn, comprises three lattices with connections between them.

The most basic lattice comprises "mental agents" or P-nodes. Some of these agents participate in any memorable event - an experience, an idea, or a problem solution - and become associated with that event. Reactivation of some of those agents recreates a "partial mental state" resembling the original.

Reactivation of P-nodes and consequent recreation of partial mental states is performed by K-Lines attached to nodes. The K-Lines originate in K-nodes embedded in a second lattice, the K-Pyramid. The establishment of a K-Line between a K-node and some P-nodes occurs when a memorable mental event takes place. Information in the K-Pyramid flows downward, but not upward.

The third structure in Minsky's model is the N-Pyramid. Its function is to permit learning to take place, and it does so by constructing new K-nodes for P.

It may be useful to think of this structure as analogous to an ivory mandarin ball - spheres within spheres.

Memory Organization Packets: MOPS (Schank, 1981)

While Minsky presents a general theory of memory without reference to the representation of concepts, Schank's model is highly dependent on the interrelationship of representations and memory. This is, perhaps, the result of the particular domain explored by Schank—temporal sequences of events.

The scripts, plans, goals, and themes of Schank's model are reminiscent of Minsky's P, K, and N pyramids, although including an additional level. An attempt to specify the relationship more precisely suggests the equivalence of scripts with the P-Pyramid, of plans with the K-Pyramid, and of goals with the N-Pyramid. The level of themes does not seem to be included in Minsky's model, although this may be a reflection of the "society of mind", with each theme representing a member of the society.

In addition to differentiating levels of description, Schank's model distinguishes four levels of memory by the degree of detail represented. These levels are:

1. Event Memory (EM) - specific remembrances of particular situations. After a while, the less salient aspects of an event fade away leaving generalized events plus the unusual or interesting parts of the original event.
2. Generalized Event Memory (GEM) - collocations of events whose common features have been abstracted. This is where general information is held about situations that have been experienced numerous times - e.g. dentists' waiting rooms.
3. Situational Memory (SM) - contains information about specific situations - e.g. going to medical practitioners' offices.
4. Intentional Memory (IM) - remembrances of generalizations at a higher level than SM - e.g.

getting a problem taken care of by an organization.

Scripts of particular situations do not exist as permanent memory structures but are reconstructed from more general, higher level structures, to aid in interpreting events as they unfold. The parts of scripts necessary for interpreting an event are retrieved from situational level memory structures referred to as Memory Organization Packets - MOPs. MOPs are collections of events that have become generalized ("mushed together" in Schank's words) and stored under them. MOPs are the means by which an appropriate episode in memory can be retrieved to aid in interpreting a specific event. Thus, connections must be established in memory from MOPs to the specific events that they are invoked to help process. These connections are evocative of Minsky's K-Lines.

A MOP aids in processing an event by virtue of the fact that it contains abstractions of similar previous events and, together with particularly memorable exceptions stored in EM, provides expectations about the development of events. Thus, a MOP is a high level script and is related to even higher-level MOPs. In Schank's example, a restaurant script is a lower level MOP with connections to more general MOPs such as social situations, contracts, and obtaining services. Like MOPs, scripts are subject to temporal precedence search, produce conceptual dependencies, and contain memories, so they may be thought of as sub-MOPs.

FUTURE TRENDS AND CONCLUSION

Although considerable effort has been spent on attempts to formalize commonsense knowledge representation, none of these has yet produced an entirely satisfactory result. Thus, there is still considerable room for work in this area

Furthermore, there has been little theoretical work on commonsense representation formalisms in recent years. The bulk of the efforts have shifted to projects utilizing various formalisms to implement commonsense knowledge bases. See "Commonsense Knowledge Representation II - Implementation" in this Encyclopedia.

REFERENCES

Brachman, R.J., Fikes, R.E. & Levesque, H.J. (1983). Krypton: a functional approach to knowledge representation. *Computer 16(10*),67-73.

Doyle, J. (1984). Expert systems without computers, or trust in artificial intelligence. *AI Magazine 5(2*),59-63.

Hayes, P. J. (1979). The logic of frames. in D. Metzing (ed.) *Frame Conceptions and Text Understanding*. de Gruyter,46-61.

Levesque, H.J. (1984). Foundations of a functional approach to knowledge representation. *Artificial Intelligence 23*,155-212.

McCarthy, J. (1980). Circumscription: a form of non-monotonic reasoning. *Artificial Intelligence 13*,27-39.

McDermott, D. & Doyle, J. (1980). Non-monotonic logic I. *Artificial Intelligence 13(1,2),*41-72.

Minsky, M. (1975). A Framework for Representing Knowledge. In P.H. Winston (Ed.) *The Psychology of Computer Vision* (pp.211-277). New York: McGraw-Hill.

Minsky, M. (1981). K-Lines: A Theory of Memory. In D.A. Norman (Ed.) *Perspectives on Cognitive Science* (pp.37-85). Norwood, NJ: Ablex.

Reiter, R. (1980). A Logic For Default Reasoning. *Artificial Intelligence (13),*81-132.

Rich, E. (1983). *Artificial Intelligence*. New York: McGraw-Hill.

Schank, R.C. (1981). Language and Memory. In D.A. Norman (Ed.) *Perspectives on Cognitive Science* (pp.105-146). Norwood, NJ: Ablex.

Waltz, D.L. (1982). Artificial Intelligence. *AI Magazine 247(4),*118-133.

Woods, W.A. (1983). What's Important About Knowledge Representation? *Computer 16(10)*,22-27.

Zadeh, L.A. (1983). Commonsense knowledge representation based on fuzzy logic. *Computer 16(10),*61-65.

KEY TERMS

Belief Maintenance Systems: Systems of logic that permit theorems to be updated as new knowledge becomes available.

Commonsense Knowledge: Knowledge of the basic facts and behaviors of the everyday world.

Dispositional Models: Representations of things or facts.

Logistic Models: Modified logics that attempt to overcome the problems of representing commonsense knowledge in the classic predicate calculus.

Monotonicity: A characteristic of logic that prevents changes to existing theorems, when new information becomes available.

Non-Monotonic Logic: A logic that attempts to overcome the restrictions of monotonicity.

Propositional Models: Descriptions of representations of things or concrete facts.

Representation Formalisms: Theoretical frameworks for representing commonsense knowledge.

ENDNOTE

1   In this article, "commonsense" is written as one word, to distinguish such knowledge from the more usual "common sense" defined in the Oxford English Dictionary as "Good sound practical sense; combined tact and readiness in dealing with the every-day affairs of life; general sagacity." Others use "commonsense" only as an adjective.

Commonsense Knowledge Representation II

Phillip Ein-Dor
Tel-Aviv University, Israel

INTRODUCTION

Early attempts to implement systems that understand commonsense knowledge did so for very restricted domains. For example, the Planes system [Waltz, 1978] knew real world facts about a fleet of airplanes and could answer questions about them put to it in English. It had, however, no behaviors, could not interpret the facts, draw inferences from them or solve problems, other than those that have to do with understanding the questions. At the other extreme, SHRDLU (Winograd, 1973) understood situations in its domain of discourse (which it perceived visually), accepted commands in natural language to perform behaviors in that domain and solved problems arising in execution of the commands; all these capabilities were restricted, however, to SHRDLU's artificial world of colored toy blocks. Thus, in implemented systems it appears that there may be a trade off between the degree of realism of the domain and the number of capabilities that can be implemented.

In the frames versus logic debate (see Commonsense Knowledge Representation I - Formalisms in this Encyclopedia), the real problem, in Israel's (1983) opinion, is not the representation formalism itself, but rather that the facts of the commonsense world have not been formulated, and this is more critical than choice of a particular formalism. A notable attempt to formulate the "facts of the commonsense world" is that of **Hayes** [1978a, 1978b, 1979] under the heading of **naïve physics**. This work employs **first-order predicate calculus** to represent commonsense knowledge of the everyday physical world. The author of this survey has undertaken a similar effort with respect to **commonsense business knowledge** (**Ein-Dor and Ginzberg** 1989). Some broader attempts to formulate commonsense knowledge bases are cited in the section Commonsense Knowledge Bases.

COMMONSENSE AND EXPERT SYSTEMS

The perception that **expert systems** are not currently sufficient for **commonsense representation** is strengthened by the conscious avoidance in that field of commonsense problems. An excellent example is the following maxim for expert system construction:

Focus on a narrow specialty area that does not involve a lot of commonsense knowledge. ... to build a system with expertise in several domains is extremely difficult, since this is likely to involve different paradigms and formalisms. (Buchanan et al., 1983)

In this sense, much of the practical work on expert systems has deviated from the tradition in Artificial Intelligence research of striving for **generality**, an effort well exemplified by the General Problem Solver (Ernst and Newell, 1969) and by work in **natural language processing**. Common sense research, on the other hand, seems to fit squarely into the AI tradition for, to the attributes of common sense (Commonsense Knowledge Representation I), it is necessary to add one more implicit attribute, namely the ability to apply any commonsense knowledge in ANY relevant domain. This need for generality appears to be one of the greatest difficulties in representing common sense.

Consider, for example, commonsense information about measurement; knowledge of appropriate measures, conversions between them, and the duration of their applicability are necessary in fields as diverse as medicine, business, and physics. However, each expert system represents knowledge, including the necessary knowledge about measuring scales, in the manner most convenient for its specific purposes. No such representation is likely to be very useful in any other system in the same domain, and certainly not for systems in other domains. Thus, it appears that the reason for the inability of expert systems as currently developed to

represent general purpose common sense is primarily a function of the **generality** of commonsense versus the **specificity** of expert systems.

From a positive point of view, one of the major aims of commonsense systems must be to represent knowledge in such a way that it can be useful in any domain; i.e. when storage strategies cannot be based on prior information about the uses to which the knowledge will be put.

This, then, is the major difference between expert systems and commonsense systems; while the former deal mainly with the particular knowledge and behaviors of a strictly bounded activity, common sense must deal with all areas of knowledge and behavior not specifically claimed by a body of experts. An expert system that knows about internal medicine does not know about skin diseases or toxicology and certainly not about drilling rigs or coal mining. Common sense systems, on the other hand, should know about colds and headaches and cars and the weather and supermarkets and restaurants and "chalk and cheese and sealing wax and cabbages and kings" (Carroll, 1872).

COMMONSENSE KNOWLEDGE BASE IMPLEMENTATIONS

Given the importance of commonsense knowledge, and because such knowledge is necessary for a wide range of applications, a number of efforts have been made to construct universally applicable commonsense knowledge bases. Three of the most prominent are Cyc, ConceptNet, and WordNet.

Cyc

The **Cyc** project (**Lenat** et al. 1990; Lenat, 2006) was initiated in 1984 by **Douglas Lenat** who has been at its head ever since. The objective of the project was to build a knowledge base of all the commonsense knowledge necessary to understand the set of articles in an encyclopedia. As of 2005, the **knowledge base** contained about 15,000 predicates, 300,000 concepts, and 3,200,000 assertions – facts, rules of thumb and heuristics for reasoning about everyday objects and events. The project is still active and the knowledge base continues to grow.

The formalism employed in Cyc is the **predicate calculus** and assertions are entered manually. (Cycorp,

2007). OpenCyc, a freely available version of Cyc may be downloaded from http://www.opencyc.org/.

ConceptNet

ConceptNet (**Liu and Singh,** 2004) is a commonsense knowledge base and natural-language-processing toolkit that supports many practical textual-reasoning tasks. Rather than assertions being registered manually as in Cyc, in ConceptNet they are generated automatically from 700,000 sentences of the **Open Mind Common Sense Project** (**Singh,** 2002) provided by over 14,000 authors., There is a concise version with 200,000 assertions and a full version of 1.6 million assertions. ConceptNet is constructed as a **semantic net**. A freely available version of the system may be downloaded at http://web.media.mit.edu/~hugo/conceptnet/#download.

WordNet

WordNet (**Felbaum,** 1998) is described as follows (WordNet, 2007): "Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (**synsets**), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. … WordNet's structure makes it a useful tool for computational linguistics and natural language processing."

WordNet contains about 155,000 words, 118,000 synsets, and 207,000 word-sense pairs.

WordNet is available for free download at http://wordnet.princeton.edu/obtain.

FUTURE TRENDS AND CONCLUSION

Any system designed to process natural language must contain commonsense knowledge as do many other types of systems. Thus, the development of commonsense knowledge bases is sure to continue.

As a complete commonsense knowledge base must contain very large quantities of knowledge, the development of such a base is a very lengthy process that must be cumulative if it is to achieve its goal. Thus, commonsense knowledge base implementations will expand and improve over a lengthy period of time.

REFERENCES

Buchanan, B.G., Barstow, D., Bechtel, R., Bennet, J., Clancey, W., Kulikowski, C., Mitchell, T., & Waterman, D.A. (1983). Constructing an Expert System. in F. Hayes-Roth, D.A.Waterman,and D.B. Lenat (Eds). *Building Expert Systems*. Reading, Mass.: Addison-Wesley.

Cycorp. What's in Cyc? Retrieved May 30, 2007 from http://www.cyc.com/cyc/cycrandd/technology/whatiscyc_dir/whatsincyc.

Carroll, L. (1872). The Walrus and The Carpenter in *Through the Looking-Glass and What Alice Found There*. London: Macmillan

Ein-Dor, P. and Ginzberg, Y. (1989). Representing Commonsense Business Knowledge: An Initial Implementation. in L. F. Pau, J. Motiwalla, Y. H. Pao, & H. H. Teh (Eds.). *Expert Systems in Economics, Banking, and Management* (pp.417-426). Amsterdam: North-Holland.

Ernst, G. & Newell, A. GPS: A Case Study in Generality and Problem Solving. (1969). New York: Academic Press.

Fellbaum, C. (Ed.). (1998). WordNet: An Electronic Lexical Database. Cambridge, Mass.: The MIT Press

Hayes, P.J. (1978a). Naive Physics 1: Ontology for Liquids. University of Essex, Working Paper.

Hayes, P.J. (1978b) Naive Physics 2: Histories. Geneva: Institute for Semantic and Cognitive Studies, Working Paper.

Hayes, P.J. (1979). The Naive Physics Manifesto. in D. Michie (Ed.) *Expert Systems in the Micro Electronic Age* (pp.242-270). Edinburgh:Edinburgh University Press.

Israel, D.J. (1983) The Role of Logic in Knowledge Representation. *Computer 16, 1,*37-41.

Lenat, D. (May 30, 2006) "Computers versus Common Sense." Google TechTalks. Retrieved May 30, 2007 from http://video.google.com/videoplay?docid=-7704388615049492068..

Lenat, D., Guha, R.V., Pittman, K., Pratt, D. & Shepherd, M. (1990). CYC: Toward Programs with Common Sense. *Communications of The ACM. 33,8*, 30-49.

Liu, H. & Singh, P. (2004) ConceptNet: A Practical Commonsense Reasoning Toolkit. *BT Technology Journal 22, 4*, 211-226.

Singh, P. (2002) The Open Mind Common Sense Project. KurzweilAI.net.. Retrieved on May 30, 2007 fromhttp://www.kurzweilai.net/meme/frame.html?main=/articles/art0371.html.

Waltz, D.L. (1978). An English Language Question Answering System for A Large Relational Database. *Communications of the ACM 21,7*, 526-539.

Winograd, T. (1973). A Procedural Model of Language Understanding. in R.C. Schank & K.M. Colby (Eds) *Computer Models of Thought and Language* (pp. 152-186). San Francisco: W.H. Freeman

WordNet. "About WordNet." Retrieved June 2, 2007 from http://wordnet.princeton.edu/.

KEY TERMS

Commonsense Knowledge: Knowledge of the basic facts of the everyday world; the knowledge that any participant in a culture expects any other participant in that culture to possess.

Commonsense Knowledge Base: A knowledge base containing commonsense knowledge and mechanisms for drawing inferences or processing natural language on the basis of that knowledge.

ConceptNet: A commonsense knowledge base implementation structured as a semantic net.

Cyc: A large commonsense knowledge base implementation utilizing predicate calculus as the representation mechanism.

Expert Knowledge: Knowledge possessed by experts in a particular domain. Systems representing expert knowledge are generally rule-based.

Implementation: The construction of a computerized system to represent and manipulate commonsense knowledge.

WordNet: A commonsense knowledge base implementation based on a semantic net structure.

A Comparative Study on E-Note-Taking

Shaista Rashid
University of Bradford, UK

Dimitris Rigas
University of Bradford, UK

INTRODUCTION

In all walks of life individuals are involved in a cumulative and incremental process of knowledge acquisition. This involves the accessing, processing and understanding of information which can be gained through many different forms. These include, deliberate means by picking up a book or passive by listening to someone. The content of knowledge is translated by individuals and often recorded by the skill of note-taking, which differs in method from one person to another. This article presents an investigation into the techniques to take notes including the most popular Cornell method. A comparative analysis with the Outlining and Mapping methods are carried out stating strengths and weaknesses of each in terms of simplicity, usefulness and effectiveness. The processes of developing such skills are not easy or straightforward and performance is much influenced by cognition. Therefore, such associations regarding cognitive conceptions involve the exploration into note-taking processes encoding and storage, attention and concentration, memory and other stimuli factors such as multimedia.

The social changes within education from the traditional manner of study to electronic are being adapted by institutes. This change varies from computerising a sub-component of learning to simulating an entire lecture environment. This has enabled students to explore academia more conveniently however, is still arguable about its feasibility. The article discusses the underlying pedagogical principles, deriving instructions for the development of an e-learning environment. Furthermore, embarking on Tablet PC's to replace the blackboard in combination with annotation applications is investigated. Semantic analysis into the paradigm shift in e-learning and knowledge management replacing classroom interaction presents its potential in the learning domain. The article concludes with ideas for the design and development of an electronic note-taking platform.

BACKGROUND

Over the years, research into note-taking has been carried out intensively. The paper aims to comparatively analyse the various note-taking techniques, providing an explanation into the effectiveness and simplicity. The relationship between cognition and note-taking is studied presenting a breakdown into the processes involved. Due to the vast amount of research into cognition its relevance is imperative. Although, great research within both areas has been undertaken to design an electronic note-taking tool, an analysis into existing applications has also been conducted, with Microsoft OneNote being the most favourable. This is an annotation application that has no predefined technique to record notes or annotations and saves handwriting as an image. Throughout the literature many authors work contributing to this study will be presented.

COMPARATIVE STUDY

This article presents an insight into note-taking, the various methods, cognitive psychology and the paradigm shift from traditional manner of study to electronic.

Note-Taking Techniques

Theoretically, note-taking is perceived as the transfer of information from one mind to the other. Today, the most popular note-taking technique is the Cornell note-taking method, also referred to as 'Do-it-right-in-the-first-place'. This note-taking method was developed over 40 years ago by Professor Walter Pauk at the Cornell University (Pauk & Owens, 2005). The main purpose of developing this method was to assist students to organise their notes in a meaningful manner. This technique involves a systematic approach for arranging and condensing notes without the need to do multiple recopying. The method is simple and

effective specifying three areas only. Area A keywords, Area B notes-taking and Area C summary.

Area A is assigned to keywords or phrases, which are formed by students towards the end of the lecture. Over the years an alternative has been questions aiding recall over recognition. These cues are known to assist memory and pose as a reminder alongside helping to identify relationships, also referred as the Q-System (Pauk & Owens, 2005). Area B remains for the recording of notes during lecture. Here the student attempts to capture as much information as possible. Finally, Area C is left for the student to summarise the notes and reflect upon the main ideas of the lecture (Pauk & Owens, 2005).

The main advantage of this technique is its clear-cut and organised structure. This technique is also suitable for technical modules including Mathematics and Physics and non-technical modules such as English and History. During an engineering and applied sciences

Figure 1. The Cornell note-taking method (adapted from Pauk & Owens, 2005)

Figure 2. An example of the outlining note-taking method

workshop, experiments involving 70 student participants revealed this note-taking method is straightforward (Anderson-Rowland, Aroz, Blaisdell, Cosgrove, Fussell, McCartney & Reyes, 1996). The authors Anderson-Rowland *et al*. (1996), state this method enables the organisation of notes, entails interaction and concentration therefore; a scheduled review can be conducted immediately highlighting keywords. Moreover, as the students can summarise content this facilitates learning by increasing understanding. Additionally, the strength of the technique is the ability to take notes instantaneously, saving time and effort due to its systematic structure.

In comparison, the Outlining method (see Figure 2) has a more spatial and meaningful layout, implicitly encoding conceptual relations. For example, indentation may imply grouping and proximity conceptual closeness (Ward & Tatsukawa, 2003).

The method consists of dashes or indentation and is not suitable for technical modules such as Mathematics or Physics. This technique requires indentation with spaces towards the right for specific facts. Relationships are represented through indentation. Note-takers are required to specify points in an organised format arranging a pattern and sequence built by space indentation. Important points are separated and kept furthest to the left, bringing in more specific points towards the right. Distance from the major point indicates the level of importance. The main advantage of this technique is the neatly organised structure allowing reviewing to be conducted without any difficulty. However, the Outlining method requires the student's full concentration to achieve maximum organisation of notes. Consequently, the technique is not appropriate if the lecturer is going at a fast pace. The method has also been disapproved by Fox (1959) because of its confusing organisational structure. This is mainly due to the arrangement of numerals, capitalised letters and so forth.

In contrast to the Cornell and Outlining methods, the Mapping method (see Figure 3) is a graphical representation of the lecture content. Students are stimulated to visually determine links illustrating relationships between facts and concepts. Concept maps enable brainstorming, breakdown and representation of complex scenarios, identifying and providing solutions for flaws and summarising information. To enhance accuracy students must actively participate and initiate critical thinking. However, a drawback arguably, has been the structural organisation and relationship of
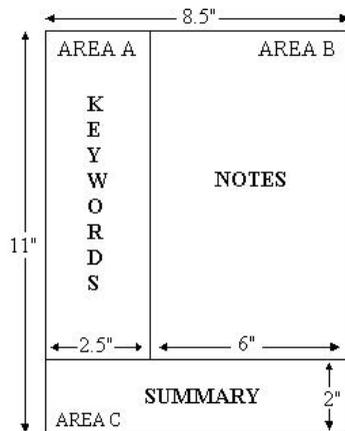
nodes especially, questioning the hierarchical structure (Hibberd, Jones, & Morris, 2002). Alternative structures have been discussed including chain, spider maps and networks (Derbentseva & Safayeni, 2004).

E-Learning

E-learning, also known as distance education employs numerous technological devices. Today, educational institutes are well-known to deliver academia over the Internet. The growth of the internet is ever-increasing with great potential for not only learning material but also as a collaborative learning environment. The major difference between traditional learning and electronic learning is the mode of instruction by which information is communicated.

To have a successful e-learning system all sub-components and interrelated processes must be considered, because if one process fails then the entire system fails. Therefore, it is necessary to derive a series of pedagogical principles. Underlying pedagogical principles include considering the user's behaviour towards the system, as it is an isolated activity resulting in user's becoming frustrated. As the internet has a great deal of knowledge it can be presented in a bias manner providing users with partial information. Considerations for the environment and the user actions to be performed to achieve a specific goal must be outlined. Moreover, user's interpersonal skills including their attitudes, perceptions and behaviour are central to influencing the effectiveness of the system. It has been learned

e-learning reduces teaching time, increase proficiency and improves retention. However, this is not always correct, as in one study, results of online lecture notes showed students performed weaker (Barnett, 2003). From student's perspective, they continuously pursue communication and support to influence their learning. They also welcome constructive feedback (Mason, 2001).

The major challenges faced by the e-learning society are the culture clash and lack of motivation towards an electronic learning environment. People are just not prepared to accept the change. A major factor determining successful outcomes of such systems is the level of interactivity provided. Other significant pedagogical principles include the level of control users feel, in comparison to traditional manner of learning where the lecturer has full control over the lecture environment. Moreover, the development of a suitable interface is necessary considering usability factors such as efficiency, error rates, memoryability, learnability and subject satisfaction. As a tutor, the planning behind the course structure is important, paying close attention towards the structure of content. Tutors must ensure students receive feedback in an appropriate time maintaining time management. Overall, before considering the design and development of an e-learning environment the main factors to consider include the learners, content, technology, teaching techniques, and so forth (Hamid, 2002).

Technologies associated with e-learning have increased usage of bandwidth and internet access. Presently, there are two key technologies used to de-

Figure 3. Example of the mapping method
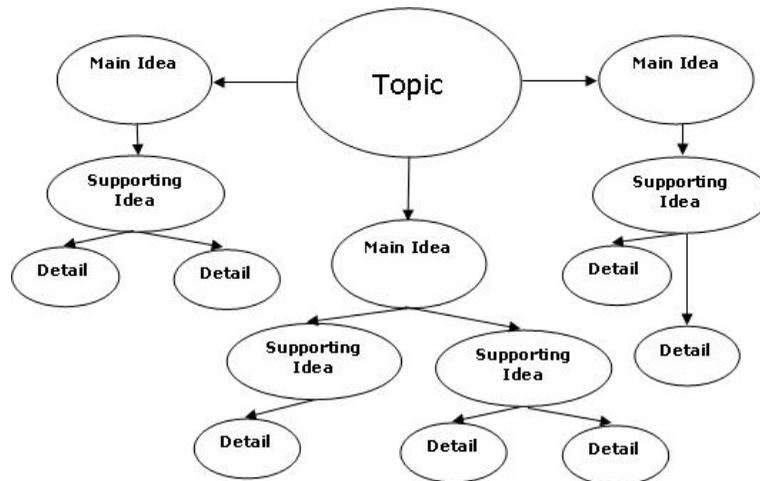
liver e-learning, 'scheduled delivery platforms' and 'on-demand delivery platforms'. Scheduled delivery platforms include multicasts, virtual libraries and remote laboratories, yet the constraints of these include time and area restraints. To further improve these systems, on-demand delivery platforms provide 24 hour support maintaining flexibility of learning (Hamilton, Richards & Sharp, 2001).

The efficacy of electronic notes in one particular study showed improvement in students understanding of lecture content, resulting in an overwhelming 96% users feeling e-notes are an effective tool (Wirth, 2003). Users are able to annotate, collaborate and discuss subject content. This increases learning efficiency by allowing the user to engage into the text, improving their comprehension and supporting memorisation. Recall of specific details is also enhanced (Wolfe, 2000). Numerous annotation applications have been introduced including Microsoft Word and OneNote primarily concerned with annotation. SharePoint™ allows manipulation editing and annotation simultaneously as well as Re:Mark™. Microsoft OneNote as an annotation tool is more popular however, the end-user is required to possess a copy of it in order to use it, unlike Microsoft Journal which can be exported as a .mhtml file (Cicchino, 2003). Additionally, handwriting is translated as an image rather then text that can be explored (McCall, 2005). The benefits include ability to record a lecture and annotate, referencing to specific points within the recording. These can then be used later (McCall, 2005).

Tablet PC's are being used to replace the blackboard presenting course material through a data projector. Many academic institutes are adopting these as a teaching tool (Clark, 2004) or are providing students with them as teaching devices (Lowe, 2004). The major strength of a Tablet PC is its interactivity between face-to-face instructions (Cicchino, 2003). However, a study that provided students with pen-based computers for the purpose of taking notes presented the project as unsuccessful because the devices where unsuitable in terms of performance, poor resolution, and network factors (Truong, Abowd & Brotherton, 1999).

Cognition

The most common mode of instruction in higher education is lectures where attending students take notes based on the lecture content (Tran & Lawson, 2001). In

class, students spend approximately 80% of their time listening to the lecture (Armbruster, 2000) and the reason students take notes is because of their usefulness towards learning and due to social pressures (Tran & Lawson, 2001). Students vary their note-taking technique according to personal experience, existing knowledge, and appropriateness to the lecture format. Problems within the classroom are caused due to student's inability to copy information presented by the lecturer (Komagata, Ohira, Kurakawa & Nakakoji, 2001). Whilst engaging in reading or listening metacognition, the human mind has a tendency to wander off thinking about thoughts other than what is being taught or learnt.

During the learning process, effects on learning occur during the encoding and storage processes. The encoding stage is when students attend lecture and record lecture notes whereas, the storage phase is when students review their notes. To achieve optimum performance, both the encoding and storage processes should be combined. The reviewing of lecture notes is significantly important especially when conducted in close proximity to an exam. Additionally, students should also monitor their individual progress and understanding of information before an exam. This can be achieved by carrying out self-testing. This method can also be encouraged by the lecturer providing relevant material for example, past exam papers.

Students with greater memory-ability benefit from note-taking with studies finding students with lower memory-ability record a lower number of words and complete ideas (Kiewra & Benton, 1988). This is due to variations within the working memory as information is stored and manipulated there. Therefore, the ability of note-takers to pick out relevant details, maintain knowledge and combine new knowledge to existing knowledge are essential factors. Human memory can be broken down into three types; Short-term; long-term and sensory memory. Short-term memory allows information to be stored for a short period before being forgotten or transferred to long-term memory. Long-term memory endures information for a longer period into the memory circuit. The brain circuit includes the cerebral cortex and hippocampus. Finally, the sensory memory is the initial storage of information lasting an instant, consisting of visual and auditory memories. Information here is typically gathered through the sight and sound senses.

The incorporation of multimedia within education can enhance the learning experience in a number of

ways. Significant sensory aids can be provided, interactivity can be increased and a richer learning experience can be initiated. The presentation of learning material and the manner in which content is captured is important. This is because if during a lecture organisational cues are explicitly defined the organisational process is guided (Titsworth & Kiewra, 2004). Moreover, the use of non-speech audio amalgamated within user-interfaces is becoming increasingly popular. If complimented with visual output it can increase the amount of information communicated to the user. A typical student captures 29% visual, 34% auditory and 37% haptic metaphors (Dryden & Vos, 1999). Sound provides greater flexibility as it can be heard 360° without having to concentrate; this is in comparison to visual output where the retina subtends an angle of 2° around the point of fixation. Consequently sound is a superb way of capturing user attention. Furthermore, graphical displays including icons, menu graphics and so forth can be used as an iconic representation of a user's action.

FUTURE TRENDS

The increase in electronic learning tools and e-learning environments within education are ever-increasing and so the necessity to derive a flexible learning structure with interactivity is paramount. Studies relating to the computerisation of note-taking tools are not so prominent. Nonetheless, the comparative study does demonstrate the effectiveness of using a note-taking method, especially the Cornell method which is most popular. Additionally, a popularity trend amongst multi-modality including audio sounds in particular earcons have shown, there is potential to combine these within learning to benefit from the experience and enhance interactivity. Moreover, in-depth research into psychological learning parameters especially encoding and storage processes have demonstrated the need to combine the two together to optimise performance. The introduction of annotation applications has instigated a trend towards electronic learning platforms although, usability issues to personalise the student experience must be further studied.

CONCLUSION

This article provides a comparative study into note-taking techniques, the processes involved and the effects of cognition upon learning. The literature suggests current electronic learning tools amalgamate many learning functions, including lecture notes, handouts, discussions and so forth into one application. Integration issues including cost factors, hardware deficiencies and malfunctions in subcomponents leading to system failures are major issues brought to light. Moreover, the primary focus into the skill and technique of note-taking have been analysed with no such technique provided as an electronic tool. Therefore, the design and development of a sub-component of e-learning, a note-taking tool is being considered based upon the studied research. The Cornell note-taking method has been arguably more effective with regards to simplicity, ability to deploy it within any subject area and appropriateness. Therefore future work will consider the influences of multimedia upon this technique, including earcons, visual structure, layout, and possibly the incorporation of speech.

REFERENCES

Anderson-Rowland, M. R., Aroz, M., Blaisdell, S., Cosgrove, C. R., Fussell, P., McCartney, M. A., & Reyes, M. (1996). Off to a Good Start: A Short, Comprehensive Orientation Program. *ASEE Annual Conference Proceedings*.

Armbruster, B.B. (2000). *Taking notes from lectures*. In R.F. Flippo & D.C Caverly (Eds.), Handbook of college reading and study strategy research, 175-199. Mahwah, NJ: Erlbaum.

Barnett, J. E. (2003). Do Instructor-Provided Online Notes Facilitate Student Learning?, *Journal of Interactive Online Learning*, 2, No.2.

Cicchino, R. M., & Mirliss, D. S. (2003). Tablet PCs: A Powerful Teaching Tool.

Clark, C. & Keating, B. (2004). Notre Dame Tablet PC Initiative. *Presented to Teaching and Learning with Technology Conference*, Purdue.

Derbentseva, N., & Safayeni, F. (2004). Experiments on the effects of map structure and concept quantifica-

C

tion during concept map construction. Concept Maps: Theory, Methodology, Technology, *Proceedings of the First International Conference on Concept Mapping*, Spain.

Dryden, G., & Vos, J. (1999). The Learning Revolution, The Learning Web, Torrance, CA, USA.

Edward W. Fox. (1959). *Syllabus for History*, Ithaca, NY: Cornell University Press.

Hamid, A. A. (2002). e-Learning Is it the "e" or the learning that matters?. *Internet and Higher Education, 4*, 311-316

Hamilton, R., Richards, C., & Sharp, C. (2001). *An Examination of E-Learning and E-Books*. available at: www.dcs.napier.ac.uk.

Hibberd, R., Jones, A., & Morris, E. (2002). The use of Concept Mapping as a Means to Promote & Assess Knowledge Acquisition, *CALRG Report No. 202*.

Kiewra, K. A., & Benton, S. L. (1988). The relationship between information-processing ability and notetaking. *Contemporary Educational Psychology, 13*, 33-44.

Komagata, N., Ohira, M., Kurakawa, K., Nakakoji, K. (2001). A cognitive system that supports note-taking in a real time class lecture. In 9th Human Interface Workshop Notes (SIG-HI-94-7). Information Processing Society of Japan, 35-40.

Lowe, P. (March 2004). Bentley College students evaluate Tablet PCs. Retrieved 3/10/2004 from http://www.hp.com/hpinfo/newsroom/feature_stories/2004/04bentley.html.

Mason, R. (2001). Time is the New Distance? An Inaugural Lecture, The Open University, Milton Keynes.

McCall, K. (2005). Digital Tools for the Digital Classroom: Digital Pens, Tablet Mice, and Tablet PCs. *20th Annual Conference on Distance Teaching and Learning*.

Monty, M.L. (1990). *Issues for Supporting notetaking and note using in the computer environment*. Ph.D. Thesis, University of California, San Diego.

Pauk, W., & Ross, O. (2005). *How to study in college*. Eighth Ed. Houghton Mifflin, 208.

Titsworth, S. B., & Kiewra, A. A. (2004). Spoken organizational lecture cues and student notetaking as facilitators of student learner. *Contemporary Educational Psychology 29*, 447-461.

Tran, T. A. T., & Lawson, M. (2001). Students' perception for reviewing lecture notes, *International Education Journal*, 2. No.4. Educational Research Conference 2001 Special Issue.

Truong, K. N., Abowd, G. D., & Brotherton, J.A. (1999). Personalizing the capture of public experiences. *In: UIST*, Ashville, NC, 121-130.

Van Meter, P., Yokoi, L., & Pressley, M. (1994). College Students' theory of note-taking derived from their perceptions of note-taking. *Journal of Educational Psychology 86*, 323-338.

Ward, N., Tatsukawa, H. (2003). A tool for taking class notes, University of Tokyo, Tokyo, Japan

Wirth, M. A. (2003). E-notes: Using Electronic Lecture Notes to Support Active Learning in Computer Science. *The SIGCSE Bulletin,* 35, No.2, 57-60.

Wolfe, J. L. (2000). Effects of Annotations on Student Readers and Writers, *Digital Libraries*, San Antonio, TX, ACM.

KEY TERMS

Annotation: The activity of briefly describing or explaining information. It can also involve summarising or evaluating content.

Cognitive Psychology: A study into cognition such as mental processes describing human behaviour, understanding perceptions, examining memory, attention span, concentration, and forgetfulness. The purpose of understanding humans and the way they mentally function.

Information Processing: The ability to capture, store and manipulate information. This consists of two main processes; encoding and storage. Students record notes during the encoding stage and conduct reviewing thereafter, in the storage phase.

Metacognition: The ability and skills of learners to be aware of and monitor their learning processes.

Multimodality: An electronic system that enhances interactivity by amalgamating audio, visual and speech metaphors.

Pedagogical Principles: Key issues to instruct the design and development of an electronic learning environment.

Platform: Computer framework allowing software to run for a specific purpose.

Traditional Manner of Study: Typically a classroom environment with the tutor writing content on a blackboard and students using pen and paper to record the content as their own notes. The tutor dominates the classroom environment unlike in electronic learning where, the user has a sense of control due to flexibility in learning.

A Comparison of Cooling Schedules for Simulated Annealing

José Fernando Díaz Martín
University of Deusto, Spain

Jesús M. Riaño Sierra
University of Deusto, Spain

INTRODUCTION

Simulated annealing is one of the most important **metaheuristics** or general-purpose algorithms of **combinatorial optimization**, whose properties of convergence towards high quality solutions are well known, although with a high computational cost. Due to that, it has been produced a quite number of research works on the convergence speed of the algorithm, especially on the treatment of the temperature parameter, which is known as **cooling schedule** or strategy. In this article we make a comparative study of the performance of simulated annealing using the most important cooling strategies (Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P., 1983), (Dowsland, K.A., 2001), (Luke, B.T., 1995), (Locatelli, M., 2000). Two classical problems of combinatorial optimization are used in the practical analysis of the algorithm: the **travelling salesman problem** and the **quadratic assignment problem**.

BACKGROUND

The main aim of **combinatorial optimization** is the analysis and the algorithmic solving of constrained optimization problems with discrete variables. Problems that require algorithms of non-polynomial time complexity with respect to the problem size, called NP-complete problems, are the most important ones.

The general solving techniques of this type of problems belong to three different, but related, research fields. First, we can mention heuristic search algorithms, such as the deterministic algorithms of *local search* (Johnson, D.S., Papadimitriou, C.H. & Yannakakis, M., 1985), (Aarts, E.H.L. & Lenstra, J., 1997), the stochastic algorithm of *simulated annealing* (Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P., 1983), and the *taboo search* (Glover, F., 1986). A second

kind of solving techniques are algorithms inspired in genetics and the evolution theory, such as *genetic and evolutionary algorithms* (Holland, J.H., 1973), (Goldberg, D.E., 1989), and *memetic algorithms* (Moscato, P., 1999). Finally, due to the collective computation properties of some neural models, the area of artificial neural networks has contributed a third approach, although possibly not so relevant as the former ones, to the combinatorial optimization problem solving with the *Hopfield nets* (Hopfield, J.J. & Tank, D., 1985), the *Boltzmann machine* (Aarts, E.H.L. & Korst, J., 1989), and the *self-organizing map* (Kohonen, T., 1988).

Simulated Annealing Algorithm

The **simulated annealing** is a stochastic variant of the **local search** that incorporates a stochastic criterion of acceptance of worse quality solutions, in order to prevent the algorithm from being prematurely trapped in local optima. This acceptance criterion is based on the **Metropolis algorithm** for simulation of physical systems subject to a heat source (Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E., 1953).

Algorithm (simulated annealing). Be a combinatorial optimization problem (X,S,f,R), with generator function of random k-neighbour feasible solutions $g : S \times [0,1[\to S$. Supposing, without loss of generality, that f must be minimized, the **simulated annealing** algorithm can be described in the following way:

1. Set an initial random feasible solution as current solution, $s_i = s_0$.
2. Set initial temperature or control parameter $T = T_0$.
3. Obtain a new solution that differs from the current one in the value of k variables using the generator

function of random k-neighbour feasible solutions, $s_j = g(s_i, random[0,1[)$.

4. If the new solution s_j is better than the current one, $f(s_j) < f(s_i)$, then s_j is set as current solution, $s_i = s_j$. Otherwise, if

$$e^{\frac{f(s_i) - f(s_j)}{T}} > random[0,1[$$

s_j is equally accepted as current solution, $s_i = s_j$.

5. If the number of executed state transitions (steps 3 and 4) for the current value of temperature is equal to L, then the temperature T is decreased.

6. If there are some k-neighbour feasible solutions near to the current one that have not been processed yet, steps 3 to 5 must be repeated. The algorithm ends in case the set of k-neighbour solutions near to the current one has been processed completely with a probability close to 1 without obtaining any improvement in the quality of the solutions.

The most important feature of the simulated annealing algorithm is that, besides accepting transitions that imply an improvement in the solution cost, it also allows to accept a decreasing number of transitions that mean a quality loss of the solution.

The simulated annealing algorithm converges asymptotically towards the set of global optimal solutions of the problem. E. Aarts and J. Korst provide a complete proof in their book *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing* (1989). Essentially, the convergence condition towards global optimum sets that temperature T of the system must be decreased logarithmically according to the equation:

$$T_k = \frac{T_0}{1 + Log(1 + k)} \qquad (1)$$

where k = 0,1,...,n indicates the temperature cycle. However, this function of system cooling requires a prohibitive computing time, so it is necessary to consider faster methods of temperature decrease.

Cooling Schedules

A practical simulated annealing implementation requires generating a finite sequence of decreasing values of temperature T, and a finite number L of state transitions for each temperature value. To achieve this aim, a ***cooling schedule*** must be specified.

The following cooling schedule, frequently used in the literature, was proposed by Kirkpatrick, Gelatt and Vecchi (1983), and it consists of three parameters:

- *Initial temperature*, T_0. The initial value of temperature must be high enough so that any new solution generated in a state transition should be accepted with a certain probability close to 1.

- *Temperature decrease function*. Generally, an exponential decrease function is used, such as $T_k = T_0 \cdot \alpha^k$, where α is a constant smaller than the unit. Usual values of α fluctuate between 0.8 and 0.99.

- *Number of state transitions*, L, for each temperature value. Intuitively, the number of transitions for each temperature must be high enough so that, if no solution changes were accepted, the whole set of k-neighbour feasible solutions near to the current one could be gone round with a probability close to 1.

The initial temperature, T_0, and the number of state transitions, L, can be easily obtained. On the other hand, the temperature decrease function has been studied in numerous research works (Laarhoven, P.J.M. Van & Aarts, E.H.L., 1987), (Dowsland, K.A., 2001), (Luke, B.T., 2005), (Locatelli, M., 2000).

MAIN FOCUS OF THE CHAPTER

In this section nine different cooling schedules used in the comparison of the simulated annealing algorithm are described. They all consist of, at least, three parameters: initial temperature T_0, temperature decrease function, and number of state transitions L for each temperature.

Multiplicative Monotonic Cooling

In the multiplicative monotonic cooling, the system temperature T at cycle *k* is computed multiplying the initial temperature T_0 by a factor that decreases with respect to cycle *k*. Four variants are considered:

- *Exponential multiplicative cooling* (Figure 1-A), proposed by Kirkpatrick, Gelatt and Vecchi (1983), and used as reference in the comparison among the different cooling criteria. The temperature decrease is made multiplying the initial temperature T_0 by a factor that decreases exponentially with respect to temperature cycle k:

$$T_k = T_0 \cdot \alpha^k \qquad (\, 0.8 \leq \alpha \leq 0.9 \,)$$
$$\text{(2)}$$

- *Logarithmical multiplicative cooling* (figure 1-B), based on the asymptotical convergence condition of simulated annealing (Aarts, E.H.L. & Korst, J., 1989), but incorporating a factor α of cooling speeding-up that makes possible its use in practice. The temperature decrease is made multiplying the initial temperature T_0 by a factor that decreases in inverse proportion to the natural logarithm of temperature cycle k:

$$T_k = \frac{T_0}{1 + \alpha Log(1+k)} \qquad (\, \alpha > 1 \,) \ \text{(3)}$$

- *Linear multiplicative cooling* (Figure 1-C). The temperature decrease is made multiplying the initial temperature T_0 by a factor that decreases in inverse proportion to the temperature cycle k:

$$T_k = \frac{T_0}{1 + \alpha k} \qquad (\, \alpha > 0 \,)$$
$$\text{(4)}$$

- *Quadratic multiplicative cooling* (Figure 1-D). The temperature decrease is made multiplying the initial temperature T_0 by a factor that decreases in inverse proportion to the square of temperature cycle k:

$$T_k = \frac{T_0}{1 + \alpha k^2} \qquad (\, \alpha > 0 \,)$$
$$\text{(5)}$$

Additive Monotonic Cooling

In the additive monotonic cooling, we must take into account two additional parameters: the number n of cooling cycles, and the final temperature T_n of the system. In this type of cooling, the system temperature T at cycle k is computed adding to the final temperature

Figure 1. Multiplicative cooling curves: (A) Exponential, (B) logarithmical, (C) linear, (D) quadratic
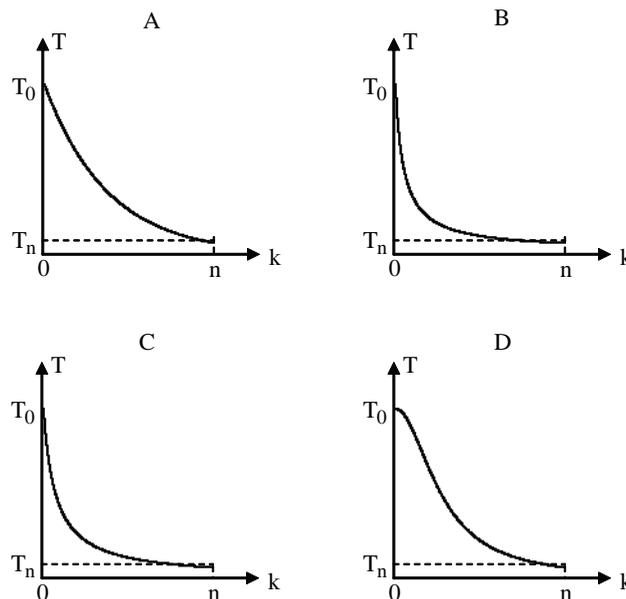
T_n a term that decreases with respect to cycle k. Four variants based on the formulae proposed by B. T. Luke (2005) are considered:

- *Linear additive cooling* (Figure 2-A). The temperature decrease is computed adding to the final temperature T_n a term that decreases linearly with respect to temperature cycle k:

$$T_k = T_n + (T_0 - T_n)\left(\frac{n-k}{n}\right)$$

(6)

- *Quadratic additive cooling* (Figure 2-B). The temperature decrease is computed adding to the final temperature T_n a term that decreases in proportion to the square of temperature cycle k:

$$T_k = T_n + (T_0 - T_n)\left(\frac{n-k}{n}\right)^2$$

(7)

- *Exponential additive cooling* (Figure 2-C). The temperature decrease is computed adding to the

final temperature T_n a term that decreases in inverse proportion to the e number raised to the power of temperature cycle k:

$$T_k = T_n + (T_0 - T_n)\left(\frac{1}{1 + e^{\frac{2\text{Ln}(T_0 - T_n)}{n}\left(k - \frac{1}{2}n\right)}}\right)$$

(8)

- *Trigonometric additive cooling* (Figure 2-D). The temperature decrease is computed adding to the final temperature T_n a term that decreases in proportion to the cosine of temperature cycle k:

$$T_k = T_n + \frac{1}{2}(T_0 - T_n)\left(1 + \cos\left(\frac{k\pi}{n}\right)\right)$$

(9)

Non-Monotonic Adaptive Cooling

In the non-monotonic adaptive cooling, the system temperature T at each state transition is computed multiplying the temperature value T_k, obtained by any of the former criteria, by an adaptive factor μ based on the difference between the current solution objective,

Figure 2: Additive cooling curves: (A) linear, (B) quadratic, (C) exponential, (D) trigonometric.
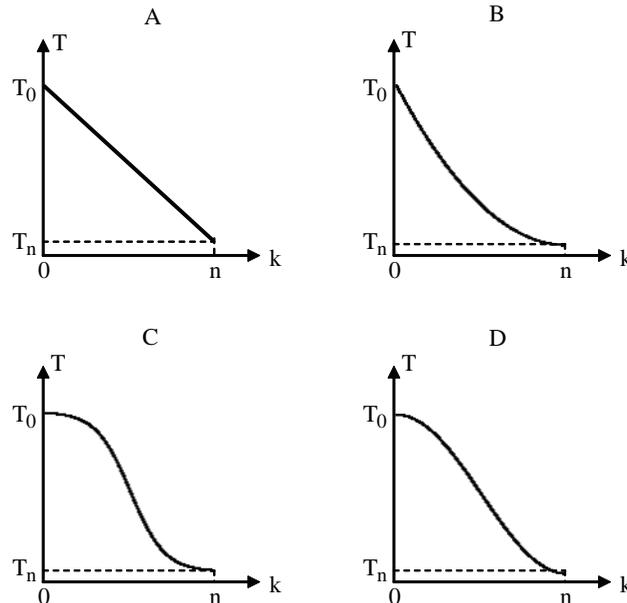
Figure 3. Curve of non-monotonic adaptive cooling
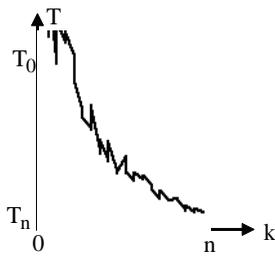
$f(s_i)$, and the best objective achieved until that moment by the algorithm, noted f^*:

$$T = \mu T_k = \left(1 + \frac{f(s_i) - f^*}{f(s_i)}\right) T_k$$

(10)

Note that the inequality $1 \le \mu < 2$ is verified. This factor μ means that the greater the distance between current solution and best achieved solution is, the greater the temperature is, and consequently the allowed energy hops. This criterion is a variant of the one proposed by M. Locatelli (2000), and it can be used in combination with any of the former criteria to compute T_k. In the comparison, the standard exponential multiplicative cooling has been used for this purpose. So the cooling curve is characterized by a fluctuant random behaviour comprised between the exponential curve defined by T_k and its double value $2T_k$ (Figure 3).

Combinatorial Optimization Problems Used in the Comparison

Travelling Salesman Problem. The Travelling Salesman Problem, TSP, consists of finding the shortest cyclic path to travel round n cities so that each city is visited only once. The objective function f to minimize is given by the expression:

$$f(s) = \sum_{i=1}^{n} d\left(x_i, x_{(i \bmod n)+1}\right),$$

where each variable x_i means the city that is visited at position i of the tour, and $d(x_i, x_j)$ is the distance between the cities x_i and x_j. The tests have been made using two

different instances of the euclidean TSP. The first one obtains a tour of 47 European cities and the second one a tour of 80 cities.

Quadratic Assignment Problem. The Quadratic Assignment Problem, QAP, consists of finding the optimal location of n workshops in p available places ($p \ge n$), considering that between each two shops a specific amount of goods must be transported with a cost per unit that is different depending on where the shops are. The objective is minimizing the total cost of goods transport among the workshops. The objective function f to minimize is given by the expression:

$$f(s) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c\left(x_i, x_j\right) q(i, j),$$

where each variable x_i means the place in which workshop i is located, $c(x_i, x_j)$ is the cost per unit of goods transport between the places where shops i and j are, and $q(i,j)$ is the amount of goods that must be transported between these shops. The tests have been made using two different instances of the QAP: The first one with 47 workshops to be located in 47 European cities, and the second one with 80 workshops to be located in 80 cities.

Selection of parameters

For each problem instance, all variants of the algorithm use the same values of initial temperature T_0 and number L of state transitions for each temperature. The initial temperature T_0 must be high enough to accept any state transition to a worse solution. The number L of state transitions must guarantee with a probability η close to 1 that, if no solution changes are accepted, any k-neighbour solution near to the current one could be process.

In order to determine the other temperature decrease parameters of each cooling schedule under similar conditions of execution time, we consider the mean final temperature $\overline{T}$ and the temperature standard error σ of the exponential multiplicative cooling of Kirkpatrick, Gelatt and Vecchi with decreasing factor $\alpha = 0.95$. The objective is to determine the temperature decrease parameters in such a way that a temperature in the interval $[\overline{T} - \sigma, \overline{T} + \sigma]$ is reached in the same number of cycles as the exponential multiplicative cooling. We distinguish three cases:

- *Multiplicative monotonic cooling*. The decrease factor α that appears in the temperature decrease equations must allow to reach the temperature $\overline{T} + \sigma$, or highest temperature from which the end of the algorithm is very probable, in the same number *n* of cycles as the exponential multiplicative cooling. Knowing that for this cooling the number *n* of cycles is:

$$\overline{T} + \sigma = T_0 \cdot (0.95)^n \Rightarrow n = \frac{\text{Log}\left(\dfrac{\overline{T} + \sigma}{T_0}\right)}{\text{Log}(0.95)}$$

(11)

it results for the logarithmic multiplicative cooling:

$$\overline{T} + \sigma = \frac{T_0}{1 + \alpha \text{Log}(1+n)} \Rightarrow \alpha = \frac{\dfrac{T_0}{\overline{T} + \sigma} - 1}{\text{Log}(1+n)}$$

(12)

for the linear multiplicative cooling:

$$\overline{T} + \sigma = \frac{T_0}{1 + \alpha n} \Rightarrow \alpha = \frac{\dfrac{T_0}{\overline{T} + \sigma} - 1}{n}$$

(13)

and for the quadratic multiplicative cooling:

$$\overline{T} + \sigma = \frac{T_0}{1 + \alpha n^2} \Rightarrow \alpha = \frac{\dfrac{T_0}{\overline{T} + \sigma} - 1}{n^2} \quad (14)$$

- *Additive monotonic cooling*. Final temperature is $T_n = \overline{T} - \sigma$, and number *n* of temperature cycles is equal to the corresponding number of cycles of the exponential multiplicative cooling for that final temperature, that is:

$$\overline{T} - \sigma = T_0 \cdot (0.95)^n \Rightarrow n = \frac{\text{Log}\left(\dfrac{\overline{T} - \sigma}{T_0}\right)}{\text{Log}(0.95)}$$

(15)

- *Non-monotonic adaptive cooling*. As the adaptive cooling is combined in the tests with the exponential multiplicative cooling, the decrease factor α that appears in the temperature decrease equation must be also $\alpha = 0.95$.

Analysis of Results

Table 1 shows the cooling parameters used on each instance of the TSP and QAP problems.

For each instance of the problems 100 runs have been made with the nine cooling schedules, computing minimum, maximum and average values, and standard error both of the objective function and of the number of iterations. For limited space reasons we only provide results for the QAP 80-workshops instance (Table 2). Local search results are also included as reference.

Considering both objective quality and number of iterations, we can conclude that the best cooling schedule we have studied is the non-monotonic adaptive cooling schedule based on the one proposed by M. Locatelli (2000), although without significant differences with respect to the exponential multiplicative and quadratic multiplicative cooling schedules.

FUTURE TRENDS

Complying with the former results we propose some research continuation lines on simulated annealing cooling schedules:

- Analyse *the influence of the initial temperature in the performance of the algorithm*. An interesting idea could be valuing the algorithm behaviour when the temperature is initialized with a percentage between 10% and 90% of the usual estimated value for T_0.
- Determine *an optimal monotonic temperature decrease curve*, based on the exponential multiplicative cooling. A possibility could be changing the parameter α dynamically in the temperature decrease equation, with lower initial values ($\alpha = 0.8$) and final values closer to 1 ($\alpha = 0.9$), but achieving an average number of temperature cycles equal to the standard case with constant parameter $\alpha = 0.95$.
- Study *new non-monotonic temperature decrease methods*, combined with the exponential multiplicative cooling. These methods could be

349

Table 1. Parameters of the cooling schedules

	TSP 47 $T_0 = 18000$ $L = 3384$	**TSP 80** $T_0 = 110000$ $L = 9720$	**QAP 47** $T_0 = 3000000$ $L = 3384$	**QAP 80** $T_0 = 25000000$ $L = 9720$
Exp M SA	$\alpha = 0.95$	$\alpha = 0.95$	$\alpha = 0.95$	$\alpha = 0.95$
Log M SA	$\alpha = 267.24$	$\alpha = 2307.6$	$\alpha = 657.19$	$\alpha = 1576.64$
Lin M SA	$\alpha = 9.45$	$\alpha = 65.76$	$\alpha = 21.08$	$\alpha = 46.37$
Qua M SA	$\alpha = 0.0675$	$\alpha = 0.3593$	$\alpha = 0.13344$	$\alpha = 0.2635$
Additive SA	$n = 156$ $T_n = 6.06$	$n = 209$ $T_n = 2.46$	$n = 181$ $T_n = 276.5$	$n = 197$ $T_n = 1023$
Adapt SA	$\alpha = 0.95$	$\alpha = 0.95$	$\alpha = 0.95$	$\alpha = 0.95$

Table 2. Results for QAP 80

		Min	**Max**	**Average**	**Std. Error**
LS	**Objective**	251313664	254639411	252752757.8	716436.9
	Iterations	27521	81118	46850.9	12458.1
Exp M SA	**Objective**	249876139	251490352	250525795.8	344195.3
	Iterations	684213	2047970	1791246.1	170293.2
Log M SA	**Objective**	250455151	253575468	251745332.1	666407.2
	Iterations	120248	2023073	696470.2	394540.2
Lin M SA	**Objective**	249896097	251907025	250635912.2	391761.6
	Iterations	653162	2250714	1428446.1	324475.7
Qua M SA	**Objective**	249847174	251611701	250470415.8	350861.7
	Iterations	1075019	2614896	1655619.1	294039
Lin A SA	**Objective**	250641396	253763581	251874846.8	591713.9
	Iterations	1946664	2552978	2008897.2	79027.1
Qua A SA	**Objective**	250033262	251800841	250780492.4	391747.2
	Iterations	1909770	2474159	1974620	89007.6
Exp A SA	**Objective**	249833632	251808711	250665143	379160.5
	Iterations	1799372	2939097	2080203.9	222456.5
Trig A SA	**Objective**	250053171	252148088	250908285.3	431834.7
	Iterations	1919964	2458885	1974441.2	77503.3
Adapt SA	**Objective**	249902310	251553959	250481008.4	361964.3
	Iterations	1564222	2455305	1803218.6	129649.9

based, as the Locatelli's one in the comparison, on modifying the temperature according to the distance from the current objective to a reference best objective.

Although these three lines of work are independent, it seems to be clear that the ultimate objective would be integrating the results achieved with these lines, in order to build a high quality combinatorial optimization metaheuristic based on simulated annealing.

CONCLUSION

The main conclusions that can be drawn from the comparison among the cooling schedules of the simulated annealing algorithm are:

- Considering objective quality related to the shape of the temperature decrease curve, we can affirm that simulated annealing works properly with respect to the ability of escape from local minima when the curve has a moderate slope at the initial and central parts of the processing, and softer at the final part of it, just as it occurs in the standard exponential multiplicative cooling. The specific shape (convex, sigmoid) of the curve in the initial and central parts does not seem outstanding.
- Considering execution time, the standard error of the number of iterations seems to be related to the temperature decrease curve tail at the final part of the algorithm. An inversely logarithmic tail produces a softer final temperature fall and a higher standard error, while inversely quadratic and exponential tails cancel out faster, providing the best standard error values of the algorithm.
- Considering the use of a non-monotonic temperature decrease method, we can affirm that not only the utilized criterion does not make worse the general performance of the algorithm but it seems to have a favourable effect that deserves to be taken into account and studied in greater depth.

REFERENCES

Aarts, E.H.L. & Korst, J. (1989): *Simulated Annealing and Boltmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, Chichester.

Aarts, E.H.L. & Lenstra, J. (1997): *Local Search in Combinatorial Optimisation*. Wiley, Chichester.

Dowsland, K.A.: & Díaz, A. (2001): *Diseño de Heurísticas y Fundamentos del Recocido Simulado*. Revista Iberoamericana de Inteligencia Artificial, 20: 34-52.

Glover, F. (1986): *Future Paths for Integer Programming and Links to Artificial Intelligence*. Computers and Operations Research, 5: 533-549.

Goldberg, D.E. (1989): *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Holland, J.H. (1973): *Genetic Algorithms and the Optimal Allocation of Trials*. SIAM Journal of Computing, 2: 88-105.

Hopfield, J.J. & Tank, D. (1985): *Neural Computation of Decisions in Optimization Problems*. Biological Cybernetics, 52: 141-152.

Johnson, D.S., Papadimitriou, C.H. & Yannakakis, M. (1985): *How Easy is Local Search?* Proceedings of the Annual Symposium on Foundations of Computer Science, 39-42. Los Angeles, CA.

Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P. (1983): *Optimization by Simulating Annealing*. Science, 220: 671-680.

Kohonen, T. (1988): *Self-organization and Associative Memory*. Springer Verlag, New York.

Laarhoven, P.J.M. Van & Aarts, E.H.L. (1987): *Simulated Annealing: Theory and Applications*. Reidel, Dordrecht.

Locatelli, M.: (2000) *Convergence of a Simulated Annealing Algorithm for Continuous Global Optimization*. Journal of Global Optimization, 18: 219-234.

Luke, B.T. (2005): *Simulated Annealing*. Technical Report. http://fconyx.ncifcrt.gov/~lukeb/simanfl.html.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953): *Equation of State Calculations*

C

by Fast Computing Machines. Journal of Chemical Physics, 21: 1087-1092.

Moscato, P. (1999): *Memetic Algorithms: A Short Introduction*. In D. Corne, M. Dorigo & F. Glover (eds.), *New Ideas in Optimization*, 219-234. McGraw-Hill, Maidenhead, Berkshire, England, UK.

Toda, M., Kubo, R. & Saitô, N. (1983): *Statistical Physics*. Springer-Verlag, Berlin, 1983.

KEY TERMS

Combinatorial Optimization: Area of the optimization theory whose main aim is the analysis and the algorithmic solving of constrained optimization problems with discrete variables.

Cooling Schedule: Temperature control method in the simulated annealing algorithm. It must specify the initial temperature T_0, the finite sequence of decreasing values of temperature, and the finite number L of state transitions for each temperature value.

Genetic and Evolutionary Algorithms: Genetic Algorithms (GAs) are approximate optimization algorithms inspired on genetics and the evolution theory. The search space of solutions is seen as a set of organisms grouped into populations that evolve in time by means of two basic techniques: crossover and mutation. Evolutionary Algorithms (EAs) are especial genetic algorithms that only use mutation as organism generation technique.

Local Search: Local search (LS) is a metaheuristic or general class of approximate optimization algorithms, based on the deterministic heuristic search technique called hill-climbing.

Memetic Algorithms: Memetic Algorithms (MAs) are optimization techniques based on the synergistic combination of ideas taken from other two metaheuristics: genetic algorithms and local search.

Simulated Annealing: Simulated Annealing (SA) is a variant of the metaheuristic of local search that incorporates a stochastic criterion of acceptance of worse quality solutions, in order to prevent the algorithm from being prematurely trapped in local optima.

Taboo Search: Taboo Search (TS) is a metaheuristic superimposed on another heuristic (usually local search or simulated annealing) whose aim is to avoid search cycles by forbidding or penalizing moves which take the solution to points previously visited in the solution space.

Complex Systems Modeling by Cellular Automata

Jiří Kroc
Section Computational Science, The University of Amsterdam, The Netherlands

Peter M. A. Sloot
Section Computational Science, The University of Amsterdam, The Netherlands

INTRODUCTION

In recent years, the notion of complex systems proved to be a very useful concept to define, describe, and study various natural phenomena observed in a vast number of scientific disciplines. Examples of scientific disciplines that highly benefit from this concept range from physics, mathematics, and computer science through biology and medicine as well as economy, to social sciences and psychology. Various techniques were developed to describe natural phenomena observed in these complex systems. Among these are artificial life, evolutionary computation, swarm intelligence, neural networks, parallel computing, cellular automata, and many others. In this text, we focus our attention to one of them, i.e. 'cellular automata'.

We present a truly discrete modelling universe, discrete in time, space, and state: Cellular Automata (CAs) (Sloot & Hoekstra, 2007, Kroc, 2007, Sloot, Chopard & Hoekstra, 2004). It is good to emphasize the importance of CAs in solving certain classes of problems, which are not tractable by other techniques. CAs, despite theirs simplicity, are able to describe and reproduce many complex phenomena that are closely related to processes such as self-organization and emergence, which are often observed within the above mentioned scientific disciplines.

BACKGROUND

We briefly explain the idea of complex systems and cellular automata and provide references to a number of essential publications in the field.

Complex Systems

The concept of complex systems (CSs) emerged simultaneously and often independently in various scientific disciplines (Fishwick, 2007, Bak, 1996, Resnick, 1997). This could be interpreted as an indication of their universality. Despite the diversity of those fields, there exist a number of common features within all complex systems. Typically a complex system consist of a vast number of simple and locally operating parts, which are mutually interacting and producing a global complex response. Self-organization (Bak, 1996) and emergence, often observed within complex systems, are driven by dissipation of energy and/or information.

Self-organization can be easily explained with ant-colony behavior studies where a vast number of identical processes, called ants, locally interact by physical contact or by using pheromone marked traces. There is no leader providing every ant with information or instructions what it should do. Despite the lack of such a leader or a hierarchy of leaders, ants are able to build complicated ant-colonies, feed their larvae, protect the colony, fight against other colonies, etc. All this is done automatically through a set of simple local interactions among the ants. It is well known that ants are responding on each stimuli by one out of 20 to 40 (depending on ant species) reactions, these are enough to produce the observed complexity.

Emergence is defined as the occurrence of new processes operating at a higher level of abstraction then is the level at which the local rules operate. Each level usually has its own local rules different from rules operating at other levels. An emergent, like an ant-colony, is a product of the process of emergence. There can be a whole hierarchy of emergents, e.g. as in the hu-

man body, that consists of chemicals and DNA, going through polypeptides, proteins, cellular infrastructures and cycles, further on to cells and tissues, organs, and bodies. We see that self-organization and emergence are often closely linked to one another.

Cellular Automata

Early development of CAs dates back to A. Turing, S. Ulam, and J. von Neumann. We can define CA's by four mutually interdependent parts: the lattice and its variables, the neighbourhood, and the local rules (Toffoli & Margolus, 1987, Toffoli, 1984, Vichniac, 1984, Ilachinski, 2001, Wolfram, 2002, Wolfram 1994, Sloot & Hoekstra, 2007, Kroc, 2007). This is briefly explained below.

Lattices and Networks

A lattice is created by a grid of elements, for historical reasons called cells, which can be composed in one, two, three, or higher dimensional space. The lattice is typically composed of uniform cells such as, for instance squares, hexagons or triangles in two dimensions.

CAs operating on networks and graphs represent a generalization of classical CAs, which are working on regular lattices. Networks can be random or regular. Networks can have various topologies, which are classified by the degree of regularity and randomness. A lattice of cells can be interpreted as a regular network of vertices interconnected by edges. When we leave this regularity and allow some random neighbours, more precisely, if a major part of a network is regular and a smaller fraction of it is random, then we enter the domain of small-world networks. The idea of small-world networks provides a unique tool, which allows us to capture many essential properties of naturally observed phenomena especially those linked to social networks and surprisingly to (metabolic and other) networks operating within living cells. Whereas small-world networks are a mixture of regular and random networks, pure random networks have a completely different scope of use. It is worth to mention the concept of scale-free networks, which have a connectivity that does not depend on scale anymore (Kroc, 2007, Sloot, Chopard & Hoekstra, 2004).

Variables

A CA contains an arbitrary number of discrete variables. The number and range of them are dictated by the phenomenon under study. The simplest CAs are built using only one Boolean variable in one dimension (1D), see e.g. (Wolfram, 2002). Some of such simple 1D CAs express even high complexity and are shown to be capable of the universal computation.

Neighbourhoods

The neighbourhood, which is used to evaluate a local rule, is defined by a set of neighbouring cells including the updated cell itself in the case of regular lattices, Figure 1. Neighbours with relative coordinates [i, j+1], [i-1,j], [i, j-1], [i+1, j] of the updated cell [i, j] and located on North, West, South, and East, respectively, define the so called the von Neumann neighbourhood with radius $r = 1$. The Moore neighbourhood with radius $r = 1$ contains the same cells as the von Neumann neighbourhood plus diagonal cells located at relative positions [i-1, j+1], [i-1, j-1], [i+1, j-1], [i+1, j+1], i.e. North-west, South-west, South-east, and North-east, respectively.

There are many other types of neighbourhoods possible; neighbourhoods can even be spatially or temporally non-uniform. One example is the Margolus neighbourhood, used in diffusion modelling.

The boundaries for each CA can be fixed, reflecting or periodic. Periodic boundary conditions represent infinite lattices. Periodic means that, e.g. in one dimension, the most right cell of a lattice is connected to the most left lattice cell. Fixed boundary cells are kept at predefined values. Reflecting boundary cells reflect values back to the bulk of the lattice.

Local Rules

A local rule defines the evolution of each CA. Usually; it is realized by taking all variables from all cells within the neighbourhood and by evaluation of a set of logical and/or arithmetical operations written in the form of an algorithm. The vector s of those variables is updated according to the following local rule in the case of the von Neumann neighbourhood

$$s[i,j] = f(s[i,j+1], s[i-1,j], s[i,j-1], s[i+1,j]),$$

Figure 1. Four types of neighbourhood is shown on the lattice of 5 x 5 cells: (from left) the von Neumann with r =1, and r =2, the Moore with r =1, and finally a random one

C
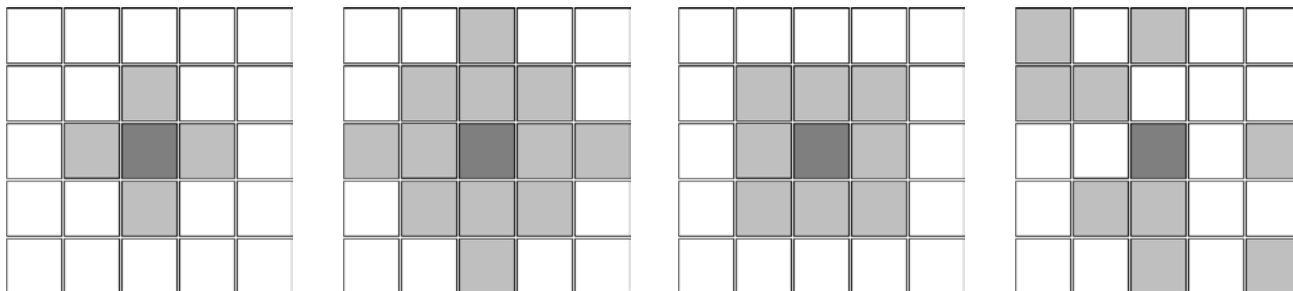
where *i* represents the x coordinate, *j* represents y co-ordinate of the cell, and *f* the local rule. The updated cell has coordinates [*i,j*]. Figure 1 shows a 5x5 two-dimensional CA with neighbourhoods having various radiuses.

Modelling

Computational modeling is defined as a mathematical, numerical and/or computational description of a naturally observed phenomenon. It is essential in situations where the observed phenomena are not tractable by analytical means. Results are often validated against analytical solutions in special or simplified cases. Its importance has been shown in physics and chemistry and is continuously increasing in new fields such as biology, medicine, sociology, and psychology.

CELLULAR AUTOMATA MODELLING OF COMPLEX SYSTEMS

There is a constant influx of new ideas and approaches enriching the CA method. Within CA modelling of complex systems, there are distinct streams of research and their applications in various disciplines, these are briefly discussed in this section.

Classical cellular automata, with a regular lattice of cells, are used to model ferromagnetic and anti-ferromagnetic materials, solidification, static and dynamic

recrystallization, laser dynamics, traffic flow, escape and pedestrian behaviour, voting processes, self-replication, self-organization, earthquakes, volcano activity, secure coding of information and cryptography, immune systems, living cells and tissue behaviour, morphological development, ecosystems, and many other natural phenomena (Sloot, Chopard & Hoekstra, 2004, Kroc, 2007, Illachinski, 2001). CAs were first used in the modelling of excitable media, such as heart tissue. CAs often outperforms other methods as, e.g., the Monte-Carlo method, especially for highly dissipative systems. The main reason why CAs represents the best choice in modelling of many naturally observed complex phenomena is because CAs are defined above truly spatio-temporally discretized worlds. The inherent CA properties brings new qualities in models that are not principally achievable by other computational techniques.

An example of an advanced CA method is the Lattice Boltzmann method consisting of a triangular network of vertices interconnected by edges where generalized 'liquid particles' move and undergo collisions according a collision table. A model of a gas is created where conservation of mass, momentum and energy during collisions are enforced, which produce a fully discrete and simplified, yet physically correct micro dynamics. When operated in the right limits, they reproduce the incompressible Navier-Stokes equations and therefore are a model for fluid dynamics. Averaged quantities resulting from such simulations correspond

to solutions of the Navier-Stokes equations (Sloot & Hoekstra, 2007, Rivet & Boon, 2001).

Classical CAs, using lattices, have many advantages over other approaches but some known disadvantages have to be mentioned. One of the disadvantages of CAs could be in the use of discrete variables. This restriction is by some authors removed by use of continuous variables, leading to generalized CAs. The biggest disadvantage of classical CAs is often found in the restricted topology of the lattice. Classical regular lattices fail to reproduce properties of many naturally observed phenomena. What led to the following development in CAs.

Generalized cellular automata, Darabos, Giacobini & Tomassini in (Kroc, 2007), are built on general networks, which are represented by regular, random, scale-free networks or small-world networks. A regular network can be created from a classical CA and its lattice where each cell represents a node and each neighbour is linked by an edge. A random graph is made from nodes that have randomly chosen nodes as neighbours. Within scale-free networks, some nodes are highly connected to other points whereas other nodes are less connected. Their properties are independent of their size. The distribution of degree of links at a node follow a power law relationship $P(k) = k^{-\gamma}$, where $P(k)$ is the probability that a node is connect to k other nodes. The coefficient γ is in most cases between 2 and 3. Those networks occur for instance in the Internet, in social networks, and in biologically produced networks such as gene regulatory networks within living cells or food chains within ecosystems (Sloot, Ivanov, Boukhanovsky, van de Vijver & Boucher, 2007).

In general, the behaviour of a given CA is unpredictable what is often used in cryptography. There exist a number of mostly statistical techniques enabling to study the behaviour of given CA but none of them is exact. The easiest way, and often the only one, to find out the state of a CA is its execution.

CASE STUDIES

Understanding morphological growth and branching of stony corals with the lattice Boltzmann method is a good example of studying natural complex system with CAs (Kaandorp, Lowe, Frenkel & Sloot, 1996, Kaandorp, Sloot, Merks, Bak, Vermeij, & Maier, 2005). A deep insight into those processes is important to assess the

role of corals in marine ecosystems and, e.g., its relation to global climate changes. Simulation of growth and branching of a coral involves multiphysics processes such as, nutrient diffusion, fluid flow, light absorption by the zooxanthele that live in symbiosis with the coral polyps, as well as mechanical stress.

It is demonstrated that nutrient gradients determine the morphogenesis of branching of phototropic corals. In this specific case, we deal with diffusion-limited processes fully determining the morphological shape of the growing corals. It is known from tank experiments and simulation studies that those diffusion dominant regions operate for relatively high flow velocities. It has been demonstrated that simulated coral morphologies are indistinguishable from real corals (Kaandorp, Sloot, Merks, Bak, Vermeij, & Maier, 2005), Figure 3.

Modelling of dynamic recrystallization represents another living application of CAs within the field of solid state physics (Kroc, 2002). Metals having polycrystalline form, composed from many single crystals, are deformed at elevated temperatures. The stored energy is increasing due to deformation, which is in turn released by recrystallization, where nuclei grow and form new grains. Growth is driven by the release of stored energy. The response of deformed polycrystalline material is reflected by complex changes within the microstructure and deformation curve.

Stress-strain curves measured during deformation of metallic samples exhibits either single peak or multiple peak behaviour. This complex response of deformed material is a direct result of concurrent processes operating within deformed material. CAs, so far, represents the only computational technique, which is able to describe such complex material behaviour (Kroc, 2002), Figure 3.

FUTURE TRENDS

There is a number of distinct tracks within CAs research with a constant flux of new discoveries (Kroc, 2007, Sloot & Hoekstra, 2007). CAs are used to model physical phenomena but they are increasingly used to model biological, medical and social phenomena. Most CAs are designed by hand but the future requires development of automatic and self-adjusting optimization techniques to design local rules according to the needs of the described natural phenomena.

C

Figure 2. Morphological growth of coral Mandracis mirabilis obtained through 3D visualization of a CT-scan of the coral (top) and two simulated growth forms (bottom) with different morphological shapes are depicted (Kaandorp, Sloot, Merks, Bak, Vermeij, & Maier, 2005). Simulated structures are indistinguishable from real corals.
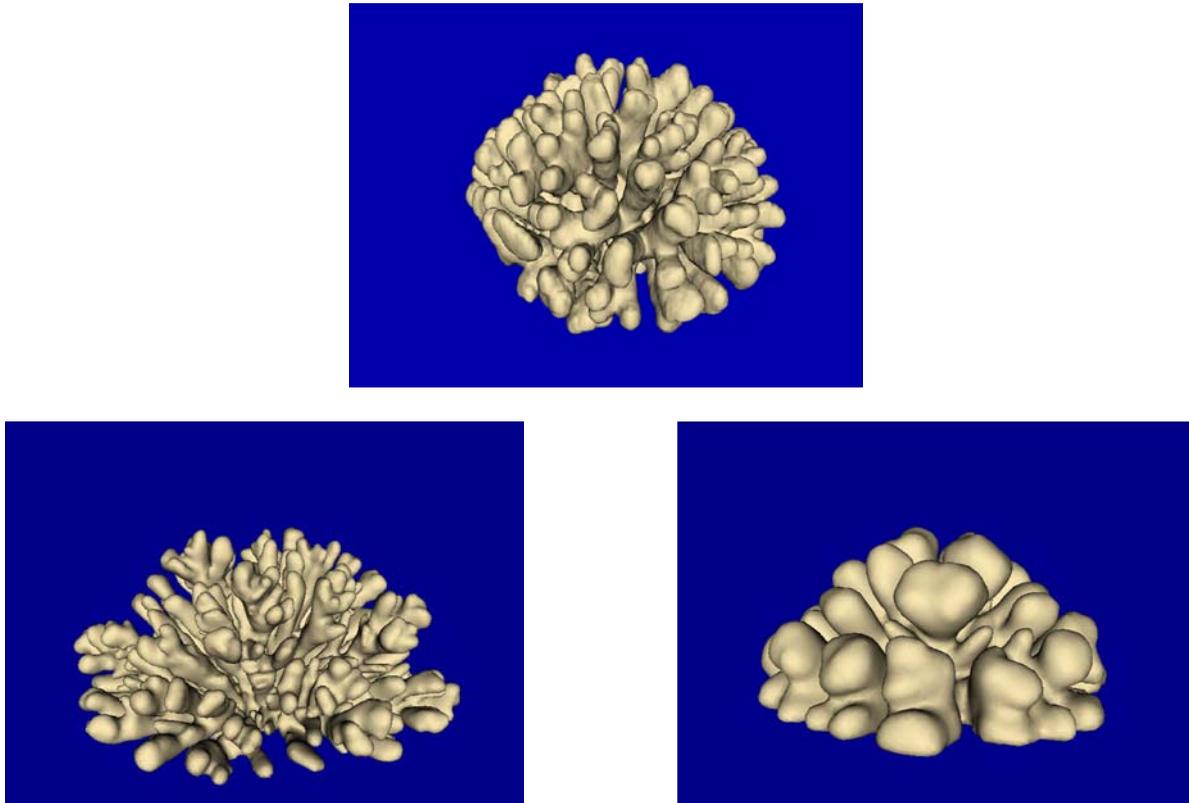
It is important to stress that the CA technique is bringing a cross-fertilization among many scientific disciplines. It happened many times in past that two or more very similar techniques were developed in distinct scientific fields such as, e.g. physics and social science.

The spatial structure of CAs is evolving from regular lattices to networked CAs Darabos, Giacobini, Tomassini in (Kroc, 2007), and to multilevel CAs (Hoekstra, Lorentz, Fakone & Chopard, 2007). Updating schemes of CAs will address in the future two regimes: synchronous (the classical one), and asynchronous (Sloot, Overeinder & Schoneveld, 2001).

CONCLUSIONS

We briefly discussed complex systems and demonstrate the usefulness of cellular automata in modelling those systems. It has been shown that cellular automata provide a simple but an extremely efficient numerical technique, which is able to describe and simulate such complicated behaviour as self-organization and emergence. This extraordinary combination of simplicity and expressivity brings a constant flux of new discoveries in description of many naturally observed phenomena in almost all scientific fields.

Figure 3. Simulation of dynamic recrystallization is represented by: stress–strain curves (top-left), relevant mean grain size D–strain curves (top-right), an abrupt change loading strain rate (bottom-left), and relevant D-strain curves (bottom-right). Strain is represented by the number of CA steps (Kroc, 2002).
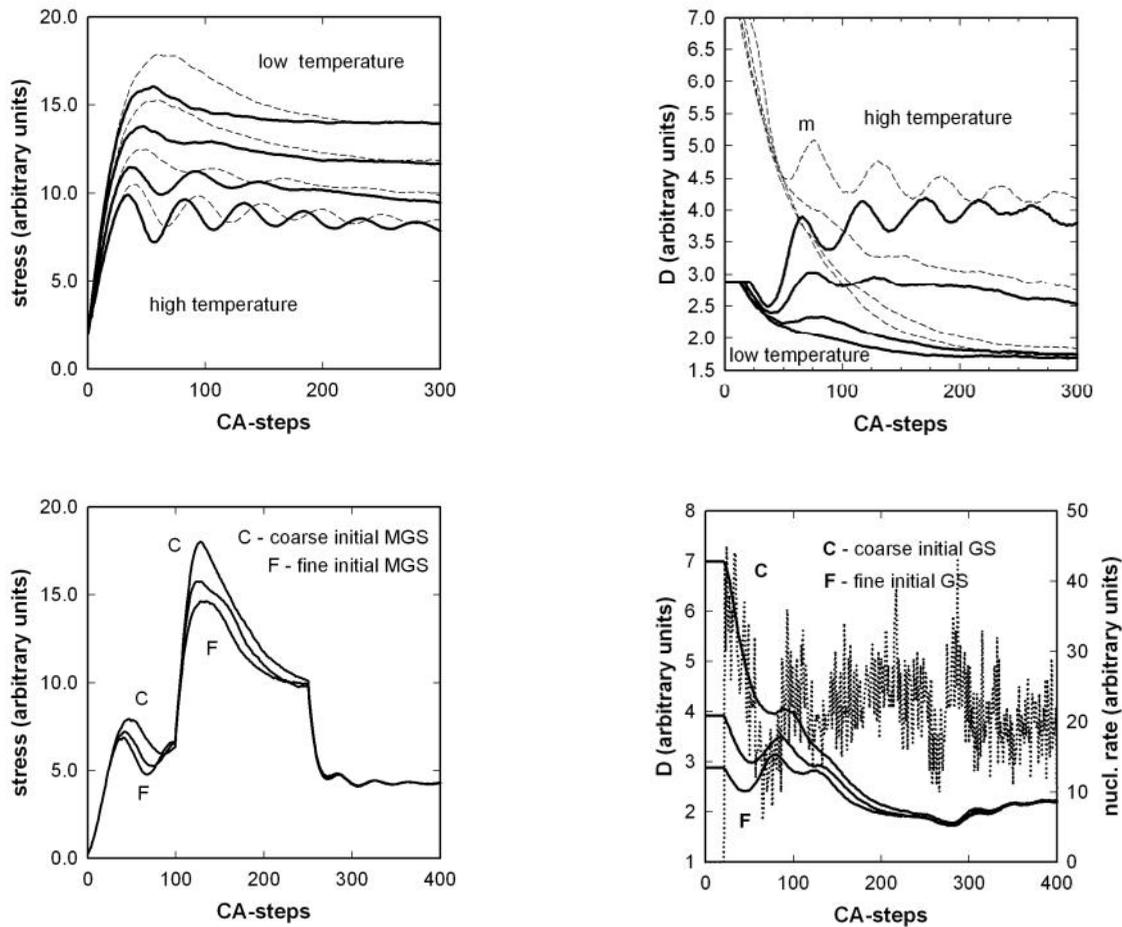
Finally, it is good to emphasize that CAs represent a generic method often used in the development of prototypes of completely new numerical methods describing naturally observed phenomena. We believe that CAs have a great potential for the future development of computational modelling and the understanding of the dynamics of complex systems.

REFERENCES

Bak, P. (1996) *How nature works: the science of self-organized criticality*. New York: Springer-Verlag Inc.

Fishwick, A. (Editor) (2007) *Handbook of Dynamical System Modeling*. Chapman & Hall/CRC Computer and Information Science Series, Taylor & Francis Group.

Hoekstra, A., Lorentz, E., Fakone, J.-L. & Chopard, B. (2007) Towards a Complex Automata Framework

C

for Multi-scale Modelling: Formalism and the Scale Separation Map, *Lecture Notes in Computer Science.* (4487) 922-930.

Illachinski, A. (2001). *Cellular Automata: A Discrete Universe.* World Scientific Publishing Co. Pte. Ltd.

Kaandorp, J.A., Lowe, C.P., Frenkel D. & Sloot P.M.A. (1996) Effect of Nutrient Diffusion and Flow on Coral Morphology, *Physical Review Letters, 77,* 2328-2331.

Kaandorp, J.A., Sloot, P.M.A., Merks, M.H., Bak, R.P.M., Vermeij, M.J.A. & Maier, C. (2005) Morphogenesis of the branching reef coral *Madracis mirabilis*, *Proc. R. Soc. B, 272*, 127-133.

Kroc, J. (2002) Application of cellular automata simulations to modeling of dynamic recrystallization, *Lecture Notes in Computer Science, 2329***, 773-782**.

Kroc, J. (Editor) (2007) *Advances in Complex Systems, 10***,** 1 supp. issue, 1-213.

Resnick, M. (1997) *Turtles, termites, and traffic jams: explorations in massively parallel microworlds.* Cambridge, Massachusetts, London, England: A Bradford Book, The MIT Press.

Rivet, J-P. & Boon, J.P. (2001) *Lattice Gas Hydrodynamics*. Cambridge University Press.

Sloot, P.M.A., Chopard, B. & Hoekstra, A.G. (Editors) (2004) *Cellular Automata: 6th international Conference on Cellular Automata for Research and Industry, ACRI 2004.* Amsterdam, The Netherlands, *LectureNotes in Computer Science* (3305) Heidelberg: Springer Verlag.

Sloot, P.M.A. & Hoekstra, A.G. (2007) Modeling Dynamic Systems with Cellular Automata, *Handbook of Dynamic System Modeling*, P.A. Fishwick editor, Chapman & Hall/CRC Computer and Information Science, Taylor & Francis Group.

Sloot, P.M.A., Ivanov, S.V., Boukhanovsky, A.V., van de Vijver, D. & Boucher, C. (2007) Stochastic Simulation of HIV Population Dynamics through Complex network Modeling. *International Journal of Computer Mathematics*. accepted.

Sloot, P.M.A., Overeinder, B.J. & Schoneveld, A. (2001) Self-organized criticality in simulated cor-

related systems. *Computer Physics Communications*, *142,* 76-81.

Toffoli, T. (1984) Cellular automata as an alternative to (rather than an approximation of) differential equations in modelling physics. *Physica D, 10***,** 117-127.

Toffoli, T. & Margolus, N. (1987) *Cellular Automata Theory*. Cambridge, Massachusetts, London, England: The MIT Press.

Vichniac G.Y. (1984) Simulation physics with cellular automata. *Physica D, 10*, 96-116.

von Neumann, J. (1951) The general and logical theory of automata. *Cerebral Mechanisms and Behavior: The Hixon Symposium*. L.A. Jeffress editor.

von Neumann, J. (1966) *Theory of Self-Reproducing Automata.* University of Illinois Press.

Wolfram, S. (2002). *A New Kind of Science*. Champaign: Wolfram Media Inc.

Wolfram, S. (1994) *Cellular Automata and Complexity. Collected Papers.* New York: Addison-Wesley Publishing Company.

KEY TERMS

Cellular Automaton: (plural: cellular automata.) A cellular automaton is defined as a lattice (network) of cells (automata) where each automaton contains a set of discrete variables, which are updated according to a local rule operating above neighbours of given cell in discrete time steps. Cellular automata are typically used as simplified but not simple models of complex systems.

Generalized Cellular Automaton: It is based on use of networks instead of regular lattices.

Complex Network: Most of biological and social networks reflect topological properties not observed within simple networks (regular, random). Two examples are small-world and scale-free networks.

Complex System: A typical complex system consists of a vast number of identical copies of several generic processes, which are operating and interacting only locally or with a limited number of not necessary

close neighbours. There is no global leader or controller associated to such systems and the resulting behaviour is usually very complex.

Emergence: Emergence is defined as the occurrence of new processes operating at a higher level of abstraction then is the level at which the local rules operate. A typical example is an ant colony where this large complex structure emerges through local interactions of ants. For example, a whole hierarchy of emergents exists and operates in a human body. An emergent is the product of an emergence process.

Lattice Gas Automata: Typically, it is a triangular network of vertices interconnected by edges where generalized liquid particles move and undergo collisions. Averaged quantities resulting from such simulations correspond to solutions of the Navier-Stokes equations.

Modelling: It is a description of naturally observed phenomena using analytical, numerical, and/or computational methods. Computational modelling is classically used in such fields as, e.g. physics, engineering. Its importance is increasing in other fields such as biology, medicine, sociology, and psychology.

Random Network: A neighbourhood of a vertex is created by a set of randomly chosen links to neighbouring vertices (elements) within a network of vertices.

Regular Lattice: A perfectly regular and uniform neighbourhood for each lattice element called cell characterizes such lattices.

Self-Organization: Self-organization is a process typically occurring within complex systems where a system is continuously fed by energy, which is transformed into a new system state or operational mode by a dissipation of energy and/or information.

Self-Organized Criticality: A complex system expressing SOC is continuously fed by energy where release of it is discrete and typically occurs in the form of avalanches. Most of its time, SOC operates at a critical point where avalanches occur. Earthquakes and volcano eruptions represent prototypical examples of SOC observed in many naturally observed phenomena.

Small-World Network: A mixture of two different types of connections within each neighbourhood characterizes small-worlds. Typically, a neighbourhood of given vertex is composed of a greater fraction of neighbours having regular short-range connectivity (regular network) and a smaller fraction of random connections (random network). Such type of neighbourhood provides unique properties to each model built on the top of it.

Complex–Valued Neural Networks

Tohru Nitta
AIST, Japan

INTRODUCTION

The usual real-valued artificial neural networks have been applied to various fields such as telecommunications, robotics, bioinformatics, image processing and speech recognition, in which complex numbers (two dimensions) are often used with the Fourier transformation. This indicates the usefulness of complex-valued neural networks whose input and output signals and parameters such as weights and thresholds are all complex numbers, which are an extension of the usual real-valued neural networks. In addition, in the human brain, an action potential may have different pulse patterns, and the distance between pulses may be different. This suggests that it is appropriate to introduce complex numbers representing phase and amplitude into neural networks.

Aizenberg, Ivaskiv, Pospelov and Hudiakov (1971) (former Soviet Union) proposed a complex-valued neuron model for the first time, and although it was only available in Russian literature, their work can now be read in English (Aizenberg, Aizenberg & Vandewalle, 2000). Prior to that time, most researchers other than Russians had assumed that the first persons to propose a complex-valued neuron were Widrow, McCool and Ball (1975). Interest in the field of neural networks started to grow around 1990, and various types of complex-valued neural network models were subsequently proposed. Since then, their characteristics have been researched, making it possible to solve some problems which could not be solved with the real-valued neuron, and to solve many complicated problems more simply and efficiently.

BACKGROUND

The generic definition of a complex-valued neuron is as follows. The input signals, weights, thresholds and output signals are all complex numbers. The net input U_n to a complex-valued neuron n is defined as:

$$U_n = \sum_m W_{nm} X_m + V_n \qquad (1)$$

where W_{nm} is the complex-valued weight connecting complex-valued neurons n and m, X_m is the complex-valued input signal from the complex-valued neuron m, and V_n is the complex-valued threshold of the neuron n. The output value of the neuron n is given by $f_C(U_n)$ where $f_C : C \to C$ is called *activation function* (C denotes the set of complex numbers). Various types of activation functions used in the complex-valued neuron have been proposed, which influence the properties of the complex-valued neuron, and a complex-valued neural network consists of such complex-valued neurons.

For example, the component-wise activation function or real-imaginary type activation function is often used (Nitta & Furuya, 1991; Benvenuto & Piazza, 1992; Nitta, 1997), which is defined as follows:

$$f_C(z) = f_R(x) + i f_R(y) \qquad (2)$$

where $f_R(u) = 1/(1+\exp(-u))$, $u \in R$ (R denotes the set of real numbers), i denotes $\sqrt{-1}$, and the net input U_n is converted into its real and imaginary parts as follows:

$$U_n = x + iy = z. \qquad (3)$$

That is, the real and imaginary parts of an output of a neuron mean the sigmoid functions of the real part x and imaginary part y of the net input z to the neuron, respectively.

Note that the component-wise activation function (eqn (2)) is bounded but non-regular as a complex-valued function because the Cauchy-Riemann equations do not hold. Here, as several researchers have pointed out (Georgiou & Koutsougeras, 1992; Nitta, 1997) in the complex region, we should recall the Liouville's

theorem, which states that if a function g is regular at all $z \in C$ and bounded, then g is a constant function. That is, we need to choose either regularity or boundedness for an activation function of complex-valued neurons. In addition, it has been proved that the complex-valued neural network with the component-wise activation function (eqn (2)) can approximate any continuous complex-valued function, whereas a network with a regular activation function (for example, $f_C(z) = 1/(1+exp(-z))$ (Kim & Guest, 1990), and $f_C(z) = tanh(z)$ (Kim & Adali, 2003)) cannot approximate any non-regular complex-valued function (Arena, Fortuna, Re & Xibilia, 1993; Arena, Fortuna, Muscato & Xibilia, 1998). That is, the complex-valued neural network with the non-regular activation function (eqn (2)) is a universal approximator, but a network with a regular activation function is not. It should be noted here that the complex-valued neural network with a regular complex-valued activation function such as $f_C(z) = tanh(z)$ with the poles can be a universal approximator on the compact subsets of the deleted neighbourhood of the poles (Kim & Adali, 2003). This fact is very important theoretically, however, unfortunately the complex-valued neural network for the analysis is not usual, that is, the output of the hidden neuron is defined as the product of several activation functions. Thus, the statement seems to be insufficient to compare with the case of component-wise complex-valued activation function. Thus, the ability of complex-valued neural networks to approximate complex-valued functions depends heavily on the regularity of activation functions used.

On the other hand, several complex-valued activation functions based on polar coordinates have been proposed. For example, Hirose (1992) proposed the following amplitude-phase type activation function:

$$f_C(z) = \tanh\left(\frac{\alpha}{m}\right) \cdot \exp(i\beta), z = \alpha \cdot \exp(i\beta),$$

(4)

where m is a constant. Although this amplitude-phase activation function is not regular, Hirose noted that the non-regularity did not cause serious problems in real applications and that the amplitude-phase framework is suitable for applications in many engineering fields such as optical information processing systems, and amplitude modulation, phase modulation and frequency modulation in electromagnetic wave communications

and radar. Aizenberg et al. (2000) proposed the following activation function:

$$f_C(z) = \exp\left(i\frac{2\pi j}{k}\right), \text{ if } \frac{2\pi j}{k} \leq \beta < \frac{2\pi(j+1)}{k},$$
$$z = \alpha \cdot \exp(i\beta), \quad j = 0, 1, \ldots, k-1$$

(5)

where k is a constant. Eqn (5) can be regarded as a type of amplitude-phase activation functions. Only phase information is used and the amplitude information is discarded, however, many successful applications show that the activation function is sufficient.

INHERENT PROPERTIES OF THE MULTI-LAYERED TYPE COMPLEX-VALUED NEURAL NETWORK

This article presents the essential differences between multi-layered type real-valued neural networks and multi-layered type complex-valued neural networks, which are very important because they expand the real application fields of the multi-layered type complex-valued neural networks. To the author's knowledge, the inherent properties of complex-valued neural networks with regular complex-valued activation functions have not been revealed except their learning performance so far. Thus, only the inherent properties of the complex-valued neural network with the non-regular complex-valued activation function (eqn (2)) are mainly described: (a) the learning performance, (b) the ability to transform geometric figures, and (c) the orthogonal decision boundary.

Learning Performance

In the applications of multi-layered type real-valued neural networks, the error back-propagation learning algorithm (called here, *Real-BP*) (Rumelhart, Hinton & Williams, 1986) has often been used. Naturally, the complex-valued version of the Real-BP (called here, *Complex-BP*) can be considered, and was actually proposed by several researchers (Kim & Guest, 1990; Nitta & Furuya, 1991; Benvenuto & Piazza, 1992; Georgiou & Koutsougeras, 1992; Nitta, 1993, 1997; Kim & Adali, 2003). This algorithm enables the network to learn complex-valued patterns naturally.

It is known that the learning speed of the Complex-BP algorithm is faster than that of the Real-BP algorithm. Nitta (1991, 1997) showed in some experiments on learning complex-valued patterns that the learning speed is several times faster than that of the conventional technique, while the space complexity (i.e., the number of learnable parameters needed) is only about half that of Real-BP. Furthermore, De Azevedo, Travessa and Argoud (2005) applied the Complex-BP algorithm of the literature (Nitta, 1997) to the recognition and classification of epileptiform patterns in EEG, in particular, dealing with spike and eye-blink patterns, and reconfirmed the superiority of the learning speed of the Complex-BP described above. As for the regular complex-valued activation function, Kim and Adali (2003) compared the learning speed of the Complex-BP using nine regular complex-valued activation functions with those of the Complex-BP using three non-regular complex-valued activation functions (including eqn (4)) through a computer simulation for a simple nonlinear system identification example. The experimental results suggested that the Complex-BP with the regular

activation function $f_C(z) = \arcsin h(z)$ was the fastest among them.

Ability to Transform Geometric Figures

The Complex-BP with the non-regular complex-valued activation function (eqn (2)) can transform geometric figures, e.g. rotation, similarity transformation and parallel displacement of straight lines, circles, etc., whereas the Real-BP cannot (Nitta, 1991, 1993, 1997). Numerical experiments suggested that the behaviour of a Complex-BP network which learned the transformation of geometric figures was related to the Identity Theorem in complex analysis.

Only an illustrative example on a *rotation* is given below. In the computer simulation, a 1-6-1 three-layered complex-valued neural network was used, which transformed a point (x, y) into (x', y') in the complex plane. Although the Complex-BP network generates a value z within the range $0 < \text{Re}[z], \text{Im}[z] < 1$ due to the activation function used (eqn (2)), for the sake of convenience it is presented in the figure given below as having a transformed value within the range $-1 < \text{Re}[z], \text{Im}[z] < 1$. The learning rate used in the experiment was 0.5. The initial real and imaginary components of the weights and the thresholds were chosen to be

random real numbers between 0 and 1. The experiment consisted of two parts: a training step, followed by a test step. The training step consisted of learning a set of (complex-valued) weights and thresholds, such that the input set of (straight line) points (indicated by black circles in Fig. 1) gave as output, the (straight line) points (indicated by white circles) rotated counterclockwise over $\pi/2$ radians. Input and output pairs were presented 1,000 times in the training step. These complex-valued weights and thresholds were then used in a test step, in which the input points lying on a straight line (indicated by black triangles in Fig. 1) would hopefully be mapped to an output set of points lying on the straight line (indicated by white triangles) rotated counterclockwise over $\pi/2$ radians. The actual output test points for the Complex-BP did, indeed, lie on the straight line (indicated by white squares). It appears that the complex-valued network has learned to generalize the transformation of each point $Z_k (= r_k \exp[i\theta_k])$ into $Z_k \exp[i\alpha] (= r_k \exp[i(\theta_k + \alpha)])$, i.e., the angle of each complex-valued point is updated by a complex-valued factor $\exp[i\alpha]$, however, the absolute length of each input point is preserved. In the above experiment, the 11 training input points lay on the line $y = -x + 1$ ($0 \leq x \leq 1$) and the 11 training output points lay on the line $y = x + 1$ ($-1 \leq x \leq 0$). The seven test input points lay on the line $y = 0.2$ ($-0.9 \leq x \leq 0.3$). The desired output test points should lie on the line $x = -0.2$.

Watanabe, Yazawa, Miyauchi and Miyauchi (1994) applied the Complex-BP in the field of computer vision. They successfully used the ability to transform geometric figures of the Complex-BP network to complement the 2D velocity vector field on an image, which was derived from a set of images and called an *optical flow*. The ability to transform the geometric figure of the Complex-BP can also be used to generate fractal images. Actually, Miura and Aiyoshi (2003) applied the Complex-BP to the generation of fractal images and showed in computer simulations that some fractal images such as snow crystals could be obtained with high accuracy where the iterated function systems (IFS) were constructed using the ability to transform geometric figure of the Complex-BP.

Orthogonal Decision Boundary

The decision boundary of the complex-valued neuron with the non-regular complex-valued activation function (eqn (2)) has a different structure from that

of the real-valued neuron. Consider a complex-valued neuron with M input neurons. Let the weights denote $\boldsymbol{w} = {}^{t}[w_{1\ldots}\ w_{M}] = \boldsymbol{w}^{r} + i\ \boldsymbol{w}^{i},\ \boldsymbol{w}^{r} = {}^{t}[w_{1}{}^{r}\ \ldots\ w_{M}{}^{r}],\ \boldsymbol{w}^{i} = {}^{t}[w_{1}{}^{i}\ \ldots\ w_{M}{}^{i}]$ and let the threshold denote $\theta = \theta^{r} + i\ \theta^{i}$. Then, for M input signals (complex numbers) $\boldsymbol{z} = {}^{t}[z_{1\ldots}\ z_{M}] = \boldsymbol{x} + i\boldsymbol{y},\ \boldsymbol{x} = {}^{t}[x_{1\ldots}\ x_{M}],\ \boldsymbol{y} = {}^{t}[y_{1\ldots}\ y_{M}]$, the complex-valued neuron generates

$$X + iY = f_{R}\left(\begin{bmatrix} {}^{t}\mathbf{w}^{r} & -{}^{t}\mathbf{w}^{i} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^{r} \right) + i\ f_{R}\left(\begin{bmatrix} {}^{t}\mathbf{w}^{i} & {}^{t}\mathbf{w}^{r} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^{i} \right)$$

(6)

as an output. Here, for any two constants $C^{R}, C^{I} \in (0, 1)$, let

$$X(\mathbf{x},\mathbf{y}) = f_{R}\left(\begin{bmatrix} {}^{t}\mathbf{w}^{r} & -{}^{t}\mathbf{w}^{i} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^{r} \right) = C^{R},$$

(7)

$$Y(\mathbf{x},\mathbf{y}) = f_{R}\left(\begin{bmatrix} {}^{t}\mathbf{w}^{i} & {}^{t}\mathbf{w}^{r} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \theta^{i} \right) = C^{I}.$$

(8)

Note here that eqn (7) is the decision boundary for the *real part* of an output of the complex-valued neuron with M inputs. That is, input signals $(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{R}^{2M}$ are classified into two decision regions $\{(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{R}^{2M} \mid X(\boldsymbol{x}, \boldsymbol{y}) \geq C^{R}\}$ and $\{(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{R}^{2M} \mid X(\boldsymbol{x}, \boldsymbol{y}) < C^{R}\}$ by the hypersurface given by eqn (7). Similarly, eqn (8) is the decision boundary for the *imaginary part*. Noting that the inner product of the normal vectors of the decision boundaries (eqns (7) and (8)) is zero, we find that the decision boundary for the real part of an output of a complex-valued neuron and that for the imaginary part intersect orthogonally.

As is well known, the XOR problem and the detection of symmetry problem cannot be solved with a single real-valued neuron (Rumelhart, Hinton & Williams, 1986). Contrary to expectation, it was proved that such problems could be solved by a single complex-valued neuron with the orthogonal decision boundary, which revealed the potent computational power of complex-valued neural networks (Nitta, 2004a).

FUTURE TRENDS

Many application results (Hirose, 2003) such as associative memories, adaptive filters, multi-user communication and radar image processing suggest directions for future research on complex-valued neural networks.

It is natural that the inherent properties of the multi-layered type complex-valued neural network with non-regular complex-valued activation function (eqn (2)) are not limited to the ones described above. Furthermore, to the author's knowledge, the inherent properties except the learning performance of recurrent type complex-valued neural networks have not been reported. The same is also true of the complex-valued neural network with regular complex-valued activation functions. Such exploration will expand the application fields of complex-valued neural networks.

In the meantime, efforts have already been made to increase the dimensionality of neural networks, for example, three dimensions (Nitta, 2006), quaternions (Arena, Fortuna, Muscato & Xibilia, 1998; Isokawa, Kusakabe, Matsui & Peper, 2003; Nitta, 2004b), Clifford algebras (Pearson & Bisset, 1992; Buchholz & Sommer, 2001), and N dimensions (Nitta, 2007), which is a new direction for enhancing the ability of neural networks.

CONCLUSION

This article outlined the inherent properties of complex-valued neural networks, especially those of the case with non-regular complex-valued activation functions, that is, (a) the learning performance, (b) the ability to transform geometric figures, and (c) the orthogonal decision boundary. Successful applications of such networks were also described.

REFERENCES

Aizenberg, I. N., Aizenberg, N. N., & Vandewalle, J. (2000). *Multi-valued and universal binary neurons*. Boston: Kluwer Academic Publishers.

Aizenberg, N. N., Ivaskiv, Yu. L., Pospelov, D. A., & Hudiakov, G. F. (1971). Multiple-valued threshold functions, boolean complex-threshold functions and

their generalization. *Kibernetika (Cybernetics)*. 4, 44-51 (in Russian).

Arena, P., Fortuna, L., Muscato, G., & Xibilia, M. G. (1998). *Neural networks in multidimensional domains. Lecture Notes in Control and Information Sciences*, 234, London: Springer.

Arena, P., Fortuna, L., Re, R., & Xibilia, M. G. (1993). On the capability of neural networks with complex neurons in complex valued functions approximation. *Proc. IEEE Int. Conf. on Circuits and Systems*, 2168-2171.

Benvenuto, N., & Piazza, F. (1992). On the complex backpropagation algorithm. *IEEE Trans. Signal Processing*, 40(4), 967-969.

Buchholz, S., & Sommer, G. (2001). Introduction to neural computation in Clifford algebra. In Sommer, G. (Ed.), *Geometric computing with Clifford algebras* (pp. 291-314). Berlin Heidelberg: Springer.

De Azevedo, F. M., Travessa, S. S., & Argoud F. I. M. (2005). The investigation of complex neural network on epileptiform pattern classification. *Proc. The 3rd European Medical and Biological Engineering Conference (EMBEC'05)*, 2800-2804.

Georgiou, G. M., & Koutsougeras, C. (1992). Complex domain backpropagation. *IEEE Trans. Circuits and Systems--II: Analog and Digital Signal Processing*, 39(5), 330-334.

Hirose, A. (1992). Continuous complex-valued back-propagation learning. *Electronics Letters,* 28(20), 1854-1855.

Hirose, A. (2003). *Complex-valued neural networks*, Singapore: World Scientific Publishing.

Isokawa, T., Kusakabe, T., Matsui, N., & Peper, F. (2003). Quaternion neural network and its application. In Palade, V., Howlett, R. J., & Jain, L. C. (Ed.), *Lecture notes in artificial intelligence, 2774 (KES2003)* (pp. 318-324), Berlin Heidelberg: Springer.

Kim, T., & Adali, T. (2003). Approximation by fully complex multilayer perceptrons. *Neural Computation*, 15(7), 1641-1666.

Kim, M. S., & Guest, C. C. (1990). Modification of backpropagation networks for complex-valued signal processing in frequency domain. *Proc. Int. Joint Conf. on Neural Networks*, 3, 27-31.

Miura, M., & Aiyoshi, E. (2003). Approximation and designing of fractal images by complex neural networks. *EEJ Trans. on Electronics, Information and Systems*, 123(8), 1465-1472 (in Japanese).

Nitta, T., & Furuya, T. (1991). A complex back-propagation learning. *Transactions of Information Processing Society of Japan*, 32(10), 1319-1329 (in Japanese).

Nitta, T. (1993). A complex numbered version of the back-propagation algorithm. *Proc. World Congress on Neural Networks*, 3, 576-579.

Nitta, T. (1997). An extension of the back-propagation algorithm to complex numbers. *Neural Networks*, 10(8), 1392-1415.

Nitta, T. (2004a). Orthogonality of decision boundaries in complex-valued neural networks. *Neural Computation*, 16(1), 73-97.

Nitta, T. (2004b). A Solution to the 4-bit parity problem with a single quaternary neuron. *Neural Information Processing – Letters and Reviews*, 5(2), 33-39.

Nitta, T. (2006). Three-dimensional vector valued neural network and its generalization ability. *Neural Information Processing – Letters and Reviews*, 10(10), 237-242.

Nitta, T. (2007). *N*-dimensional vector neuron. *Proc. IJCAI Workshop on Complex Valued Neural Networks and Neuro-Computing: Novel Methods, Applications and Implementations*, 2-7.

Pearson, J., & Bisset, D. (1992). Back propagation in a Clifford algebra. *Proc. International Conference on Artificial Neural Networks*, *Brighton*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Parallel distributed processing (Vol.1)*. MA: The MIT Press.

Watanabe, A., Yazawa, N., Miyauchi, A., & Miyauchi, M. (1994). A method to interpret 3D motions using neural networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E77-A(8), 1363-1370.

Widrow, B., McCool, J., & Ball, M. (1975). The complex LMS algorithm. *Proceedings of the IEEE*. 63(4), 719-720.

KEY TERMS

Artificial Neural Network: A network composed of artificial neurons. Artificial neural networks can be trained to find nonlinear relationships in data.

Back-Propagation Algorithm: A supervised learning technique used for training neural networks, based on minimizing the error between the actual outputs and the desired outputs.

Clifford Algebras: An associative algebra, which can be thought of as one of the possible generalizations of complex numbers and quaternions.

Complex Number: A number of the form $a + ib$ where a and b are real numbers, and i is the imaginary unit such that $i^2 = -1$. a is called the *real part*, and b the *imaginary part*.

Decision Boundary: A boundary which pattern classifiers such as the real-valued neural network use to classify input patterns into several classes. It generally consists of hypersurfaces.

Identity Theorem: A theorem for regular complex functions: given two regular functions f and g on a connected open set D, if $f = g$ on some neighborhood of z that is in D, then $f = g$ on D.

Quaternion: A four-dimensional number which is a non-commutative extension of complex numbers.

Regular Complex Function: A complex function that is complex-differentiable at every point.

Figure 1. Rotation of a straight line. A black circle denotes an input training point, a white circle an output training point, a black triangle an input test point, a white triangle a desired output test point, and a white square an output test point generated by the Complex-BP network.
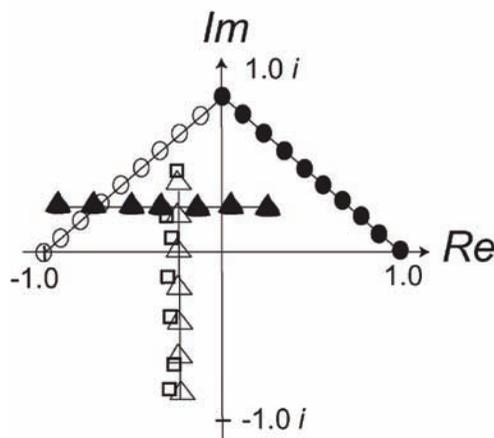
Component Analysis in Artificial Vision

Oscar Déniz Suárez
University of Las Palmas de Gran Canaria, Spain

Gloria Bueno García
University of Castilla – La Mancha, Spain

INTRODUCTION

The typical recognition/**classification** framework in Artificial Vision uses a set of object features for discrimination. Features can be either numerical measures or nominal values. Once obtained, these feature values are used to classify the object. The output of the classification is a label for the object (Mitchell, 1997).

The classifier is usually built from a set of "**training**" samples. This is a set of examples that comprise feature values and their corresponding labels. Once trained, the classifier can produce labels for new samples that are not in the training set.

Obviously, the extracted features must be discriminative. Finding a good set of features, however, may not be an easy task. Consider for example, the face recognition problem: recognize a person using the image of his/her face. This is currently a hot topic of research within the Artificial Vision community, see the surveys (Chellappa et al, 1995), (Samal & Iyengar, 1992) and (Chellappa & Zhao, 2005). In this problem, the available features are all of the pixels in the image. However, only a number of these pixels are normally useful for discrimination. Some pixels are background, hair, shoulders, etc. Even inside the head zone of the image some pixels are less useful than others. The eye zone, for example, is known to be more informative than the forehead or cheeks (Wallraven et al, 2005). This means that some features (pixels) may actually increase recognition error, for they may confuse the classifier.

Apart from performance, from a computational cost point of view it is desirable to use a minimum number of features. If fed with a large number of features, the classifier will take too long to train or classify.

BACKGROUND

Feature Selection aims at identifying the most **informative features**. Once we have a measure of "*informativeness*" for each feature, a subset of them can be used for classifying. In this case, the features remain the same, only a selection is made. The topic of feature selection has been extensively studied within the Machine Learning community (Duda et al, 2000). Alternatively, in Feature Extraction a new set of features is created from the original set. In both cases the objective is both reducing the number of available features and using the most discriminative ones.

The following sections describe two techniques for Feature Extraction: Principal Component Analysis and Independent Component Analysis. Linear Discriminant Analysis (LDA) is a similar dimensionality reduction technique that will not be covered here for space reasons, we refer the reader to the classical text (Duda et al., 2000).

Figure 1.

As an example problem we will consider face recognition. The face recognition problem is particularly interesting here because of a number of reasons. First, it is a topic of increasingly active research in Artificial Vision, with potential applications in many domains. Second, it has images as input, see Figure 1 from the Yale Face Database (Belhumeur et al, 1997), which means that some kind of feature processing/selection must be done previous to classification.

PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA), see (Turk & Pentland, 1991), is an orthogonal linear transformation of the input feature space. PCA transforms the data to a new coordinate system in which the data variances in the new dimensions is maximized. Figure 2 shows a 2-class set of samples in a 2-feature space. These data have a certain **variance** along the horizontal and vertical axes. PCA maps the samples to a new orthogonal coordinate system, shown in bold, in which the sample variances are maximized. The new coordinate system is centered on the data mean.

The new set of features (note that the coordinate axes are features) is better from a discrimination point of view, for samples of the two classes can be readily separated. Besides, the PCA transform provides an ordering of features, from the most discriminative (in terms of variance) to the least. This means that we can select and use only a subset of them. In the figure above, for example, the coordinate with the largest axis is the most discriminative.

When the input space is an image, as in face recognition, training images are stored in a matrix T. Each row of T contains a training image (the image rows are laid consecutively, forming a vector). Thus, each image pixel is considered a feature. Let there be n training images. PCA can then be done in the following steps:

1. Subtract the mean image vector m from T, where:

$$m = \frac{1}{n} \sum_i x_i$$

2. Calculate the covariance matrix C:

$$C = \frac{1}{n} \sum_i (x_i - m)(x_i - m)^{\mathrm{T}}$$

3. Perform Singular Value Decomposition over C, which gives an orthogonal transform matrix W

Figure 2.
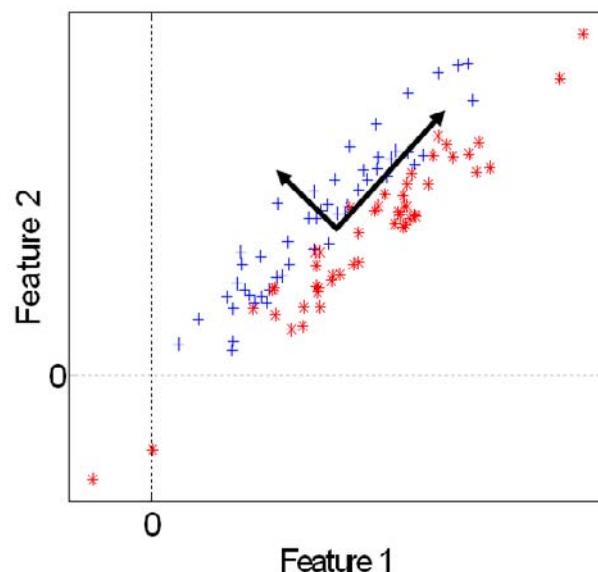
Figure 3.

4.   Choose a set of "eigenfaces" (see below)

The new feature axes are the columns of *W*. These features can be considered as images (i.e. by arranging each vector as a matrix), and are commonly called basis images or *eigenfaces* within the face recognition community. The intensity of the pixels of these images represents their weight or contribution in the axis. Figure 3 shows the typical aspect of eigenfaces.

Normally, in step 4 above only the best K eigenvectors are selected and used in the classifier. That is achieved by discarding a number of columns in *W*. Once we have the appropriate transform matrix, any set *X* of images can be transformed to this new space simply by:

1.   Subtract the mean *m* from the images in *X*
2.   Calculate $Y = X \cdot W$

The transformed image vectors *Y* are the new feature vectors that the classifier will use for training and/or classifying.

INDEPENDENT COMPONENT ANALYSIS

Independent Component Analysis is a feature extraction technique based in extracting statistically independent variables from a mixture of them (Jutten & Hérault,

1991, Comon, 1994). ICA has been successfully applied to many different problems such as MEG and EEG data analysis, **blind source separation** (i.e. separating mixtures of sound signals simultaneously picked up by several microphones), finding hidden factors in financial data and face recognition, see (Bell & Sejnowski, 1995).

The ICA technique aims at finding a linear transform for the input data so that the transformed data is as statistically independent as possible. **Statistical independence** implies **decorrelation** (but note that the opposite is not true). Therefore, ICA can be considered a generalization of PCA.

The basis images obtained with ICA are more local than those obtained with PCA, which suggests that they can lead to more precise representations. Figure 4 shows the typical basis images obtained with ICA.

ICA routines are available in a number of different implementations, particularly for Matlab. ICA has a higher computational cost than PCA. FastICA is the most efficient implementation to date, see (Gävert et al, 2005).

As opposed to PCA, ICA does not provide an intrinsic order for the representation coefficients of the face images, which does not help when extracting a subset of K features. In (Bartlett & Sejnowski, 1997) the best results were obtained with an order based on the ratio of between-class to within-class variance for each coefficient:

Figure 4.

Samal, A. & Iyengar, P.A. (1992). Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, vol. 25, no. 1.

Turk, M.A. & Pentland, A. (1991). Eigenfaces for Recognition. *Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86.

van der Heijden, F., Duin, R.P.W, de Ridder, D. & Tax, D.M.J. (2004). Classification, parameter estimation and state estimation - an engineering approach using Matlab. John Wiley & Sons. PRTools available from *http://www.prtools.org*

Wallraven, C., Schwaninger, A. & Bülthoff, H.H. (2005). Learning from Humans: Computational Modeling of Face Recognition. *Network: Computation in Neural Systems* 16(4), 401-418.

Yan, S., Xu, D., Zhang, B., Zhang, H. (2005). Graph Embedding : A General Framework for Dimensionality Reduction. *Procs. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).*

KEY TERMS

C

Classifier: Algorithm that produces class labels as output, from a set of features of an object. A classifier, for example, is used to classify certain features extracted from a face image and provide a label (an identity of the individual).

Eigenface: A basis vector of the PCA transform, when applied to face images.

Face Recognition: The AV problem of recognizing an individual from one or more images of his/her face.

Feature Extraction: The process by which a new set of discriminative features is obtained from those available. Classification is performed using the new set of features.

Feature Selection: The process by which a subset of the available features (usually the most discriminative ones) is selected for classification.

Independent Component Analysis: Feature extraction technique in which the statistical independence of the data is maximized.

Principal Component Analysis: Feature extraction technique in which the variance of the data is maximized. It provides a new feature space in which the dimensions are ordered by sample correlation. Thus, a subset of these dimensions can be chosen in which samples are minimally correlated.

Computational Methods in Biomedical Imaging

Michele Piana
Universita' di Verona, Italy

INTRODUCTION

Biomedical imaging represents a practical and conceptual revolution in the applied sciences of the last thirty years. Two basic ingredients permitted such a breakthrough: the technological development of hardware for the collection of detailed information on the organ under investigation in a less and less invasive fashion; the formulation and application of sophisticated mathematical tools for signal processing within a methodological setting of truly interdisciplinary flavor.

A typical acquisition procedure in biomedical imaging requires the probing of the biological tissue by means of some emitted, reflected or transmitted radiation. Then a mathematical model describing the image formation process is introduced and computational methods for the numerical solution of the model equations are formulated. Finally, methods based on or inspired by Artificial Intelligence (AI) frameworks like machine learning are applied to the reconstructed images in order to extract clinically helpful information.

Important issues in this research activity are the intrinsic numerical instability of the reconstruction problem, the convergence properties and the computational complexity of the image processing algorithms. Such issues will be discussed in the following with the help of several examples of notable significance in the biomedical practice.

BACKGROUND

The first breakthrough in the theory and practice of recent biomedical imaging is represented by X-ray Computerized **Tomography** (CT) (Hounsfield, 1973). On October 11 1979 Allan Cormack and Godfrey Hounsfield gained the Nobel Prize in medicine for the development of computer assisted tomography. In the press release motivating the award, the Nobel Assembly of the Karolinska Institut wrote that in this revolutionary diagnostic tool "the signals[...]are stored and mathematically analyzed in a computer. The computer is programmed to reconstruct an image of the examined cross-section by solving a large number of equations including a corresponding number of unknowns". Starting from this crucial milestone, biomedical imaging has represented a lively melting pot of clinical practice, experimental physics, computer science and applied mathematics, providing mankind of numerous non-invasive and effective instruments for early detection of diseases, and scientist of a prolific and exciting area for research activity.

The main imaging modalities in biomedicine can be grouped into two families according to the kind of information content they provide.

- **Structural imaging**: the image provides information on the anatomical features of the tissue without investigating the organic metabolism. Structural modalities are typically characterized by a notable spatial resolution but are ineffective in reconstructing the dynamical evolution of the imaging parameters. Further to X-ray CT, other examples of such approach are Fluorescence Microscopy (Rost & Oldfield, 2000), Ultrasound Tomography (Greenleaf, Gisvold & Bahn, 1982), structural Magnetic Resonance Imaging (MRI) (Haacke, Brown, Venkatesan & Thompson, 1999) and some kinds of prototypal non-linear tomographies like Microwave Tomography (Boulyshev, Souvorov, Semenov, Posukh & Sizov, 2004), Diffraction Tomography (Guo & Devaney, 2005), Electrical Impedance Tomography (Cheney, Isaacson & Newell, 1999) and Optical Tomography (Arridge, 1999).
- **Functional imaging**: during the acquisition many different sets of signals are recorded according to a precisely established temporal paradigm. The resulting images can provide information on metabolic deficiencies and functional diseases

C

but are typically characterized by a spatial reso-
lution which is lower (sometimes much lower)
than the one of anatomical imaging. Emission
tomographies like Single Photon Emission Com-
puterized Tomography (SPECT) (Duncan, 1997)
or Positron Emission Tomography (PET) (Valk,
Bailey, Townsend & Maisey, 2004) and Magnetic
Resonance Imaging in its functional setup (fMRI)
(Huettel, Song & McCarthy, 2004) are examples of
these dynamical techniques together with Electro-
and Magnetoencephalography (EEG and MEG)
(Zschocke & Speckmann, 1993; Hamalainen,
Hari, Ilmoniemi, Knuutila & Lounasmaa, 1993),
which reproduce the neural activity at a millisec-
ond time scale and in a completely non-invasive
fashion.

In all these imaging modalities the correct math-
ematical modeling of the imaging problem, the for-
mulation of computational algorithms for the solution
of the model equations and the application of image
processing algorithms for data interpretation are the
crucial steps which allow the exploitness of the visual
information from the measured raw data.

MAIN FOCUS

From a mathematical viewpoint the **inverse problem**
of synthesizing the biological information in a visual
form from the collected radiation is characterized by
a peculiar pathology.

The concept of **ill-posedness** has been introduced
by Jules Hadamard (Hadamard, 1923) to indicate math-
ematical problems whose solution does not exist for
all data, or is not unique or does not depend uniquely
on the data. In biomedical imaging this last feature has
particularly deleterious consequences: indeed, the pres-
ence of measurement noise in the raw data may produce
notable numerical instabilities in the reconstruction
when naive approaches are applied.

Most (if not all) biomedical imaging problems are
ill-posed inverse problems (Bertero & Boccacci, 1998)
whose solution is a difficult mathematical task and often
requires a notable computational effort. The first step
toward the solution is represented by an accurate model-
ing of the mathematical relation between the biological
organ to be imaged and the data provided by the imaging

device. Under the most general assumptions the model
equation is a non-linear integral equation, although, for
several devices, the non-linear imaging equation can
be reliably approximated by a linear model where the
integral kernel encodes the impulse response of the
instrument. Such linearization can be either performed
through a precise technological realization, like in MRI,
where acquisition is designed in such a way that the
data are just the Fourier Transform of the object to be
imaged; or obtained mathematically, by applying a sort
of perturbation theory to the non-linear equation, like
in diffraction tomography whose model comes from
the linearization of the scattering equation.

The second step toward image reconstruction is
given by the formulation of computational methods
for the reduction of the model equation. In the case of
linear ill-posed inverse problems, a well-established
regularization theory exists which attenuates the nu-
merical instability related to ill-posedness maintaining
the biological reliability of the reconstructed image.
Regularization theory is at the basis of most linear
imaging modalities and regularization methods can be
formulated in both a probabilistic and a deterministic
setting. Unfortunately an analogously well- established
theory does not exist in the case of non-linear imaging
problems which therefore are often addressed by means
of 'ad hoc' techniques.

Once an image has been reconstructed from the data,
a third step has to be considered, i.e. the processing of
the reconstructed images for the extraction and inter-
pretation of their information content. Three different
problems are typically addressed at this stage:

- **Edge detection** (Trucco & Verri, 1998)**.** Computer
  vision techniques are applied in order to enhance
  the regions of the image where the luminous
  intensity changes sharply.
- **Image integration** (Maintz & Viergever, 1998). In
  the clinical workflow several images of a patient
  are taken with different modalities and geometries.
  These images can be fused in an integrated model
  by recovering changes in their geometry.
- **Image segmentation** (Acton & Ray, 2007). Partial
  volume effects make the interfaces between the
  different tissues extremely fuzzy, thus complicat-
  ing the clinical interpretation of the restored im-
  ages. An automatic procedure for the partitioning
  of the image in homogeneous pixel sets and for

the classification of the segmented regions is at the basis of any Computed Aided Diagnosis and therapy (CAD) software.

AI algorithms and, above all, machine learning play a crucial role in addressing these image processing issues. In particular, as a subfield of machine learning, pattern recognition provides a sophisticated description of the data which, in medical imaging, allows to locate tumors and other pathologies, measure tissue dimensions, favor computer-aided surgery and study anatomical structures. For example, supervised approaches like backpropagation (Freeman & Skapura, 1991) or boosting (Shapire, 2003) accomplish classification tasks of the different tissues from the knowledge of previously interpreted images; while unsupervised methods like Self-Organizing Maps (SOM) (Kohonen, 2001), fuzzy clustering (De Oliveira & Pedrycz, 2007) and Expectation-Maximization (EM) (McLachlan & Krishnan, 1996) infer probabilistic information or identify clustering structures in sets of unlabeled images. From a mathematical viewpoint, several of these methods correspond more to heuristic recipes than to rigorously formulated and motivated procedures. However, since the last decade the theory of **statistical learning** (Vapnik, 1998) has appeared as the best candidate for a rigorous description of machine learning within a functional analysis framework.

FUTURE TRENDS

Among the main goals of recent biomedical imaging we point out the realization of

- microimaging techniques which allow the investigation of biological tissues of micrometric size for both diagnostic and research purposes;
- hybrid systems combining information from different modalities, possibly anatomical and functional;
- highly non-invasive diagnostic tools, where even a modest discomfort is avoided.

These goals can be accomplished only by means of an effective interplaying of hardware development and application of innovative image processing algorithms. For example, microtomography for biological samples requires the introduction of both new X-ray

tubes for data acquisition and computational methods for the reduction of beam hardening effects; electro-physiological and structural information on the brain can be collected by performing an EEG recording inside an MRI scanning but also using the structural information from MRI as a prior information in the analysis of the EEG signal accomplished in a Bayesian setting; finally, non-invasivity in colonoscopy can be obtained by utilizing the most recent acquisition design in X-ray tomography together with sophisticated softwares which allow virtual navigation within the bowel, electronic cleansing and automatic classification of cancerous and healthy tissues.

From a purely computational viewpoint, two important goals in machine learning applied to medical imaging are the development of algorithms for semi-supervised learning and for the automatic integration of genetic data with information coming from the acquired imagery.

CONCLUSION

Some aspects of recent biomedical imaging have been described from a computational science perspective. The biomedical image reconstruction problem has been discussed as an ill-posed inverse problem where the intrinsic numerical instability producing image artifacts can be reduced by applying sophisticated regularization methods. The role of image processing based on machine learning techniques has been described together with the main goals of recent biomedical imaging applications.

REFERENCES

Acton, S. T & Ray, N. (2007). Biomedical Image Analysis: Segmentation. Princeton: Morgan and Claypool.

Arridge, S. R. (1999). Optical Tomography in Medical Imaging. Inverse Problems. 15, R41-R93. Evolutionary Programming, Genetic Algorithms. Oxford University Press.

Bertero, M. & Boccacci, P. (1998). Introduction to Inverse Problems in Imaging. Bristol: IOP.

Boulyshev, A. E., Souvorov, A. E., Semenov, S. Y., Posukh, V. G. & Sizov, Y. E. (2004). Three-dimen-

sional Vector Microwave Tomography: Theory and Computational Experiments. Inverse Problem. 20, 1239-1259.

Cheney, M., Isaacson, D. & Newell, J. C. (1999). Electrical Impedance Tomography. SIAM Review. 41, 85-101.

De Oliveira, J. V. & Pedrycz, W (2007). Advances in Fuzzy Clustering and its Applications. San Francisco: Wiley.

Duncan, R. (1997). SPECT Imaging of the Brain. Amsterdam: Kluwer.

Freeman, J. A. & Skapura, D. M. (1991). Neural Network Algorithms: Applications and Programming Techniques. Redwood City: Addison-Wesley.

Greenleaf, J., Gisvold, J. J. & Bahn, R. (1982). Computed Transmission Ultrasound Tomography. Medical Progress Through Technology. 9, 165-170.

Guo, P. & Devaney, A. J. (2005). Comparison of Reconstruction Algorithms for Optical Diffraction Tomography. Journal of the Optical Society of America A. 22, 2338-2347.

Haacke, E. M., Brown, R. W., Venkatesan, R. & Thompson, M. R. (1999). Magnetic Resonance Imaging: Physical Principles and Sequence Design. San Francisco: John Wiley.

Hadamard, J. (1923). Lectures on Cauchy's Problem in Partial Differential Equations. Yale: Yale University Press.

Hamalainen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J. & Lounasmaa, O. V. (1993). Magnetoencephalography: theory, instrumentation and applications to non-invasive studies of the working human brain. Reviews of Modern Physics. 65, 413-497.

Hounsfield, G. N. (1973). Computerised Transverse Axial Scanning (Tomography). I: Description of System. British Journal of Radiology. 46, 1016-1022.

Huettel, S. A., Song, A. W. and McCarthy, G. (2004). Functional Magnetic Resonance Imaging. Sunderland: Sinauer Associates.

Kohonen, T. (2001). Self-Organizing Maps. Berlin: Springer.

Maintz, J. & Viergever, M. (1998). A Survey of Medical Imaging Registration. Medical Imaging Analysis, 2, 1-36.

McLachlan, G. & Krishnan, T. (1996). The EM Algorithm and Extensions. San Francisco: John Wiley.

Rost, F. & Oldfield R. (2000). Fluorescence Microscopy: Photography with a Microscope. Cambridge: Cambridge University Press.

Shapire, R. E. (2003). The Boosting Approach to Machine Learning: an Overview. Nonlinear Estimation and Classification, Denison, D. D., Hansen, M. H., Holmes, C., Mallik, B. & Yu, B. editors. Berlin: Springer.

Trucco, E. & Verri, A. (1998). Introductory Techniques for 3D Computer Vision. Englewood Cliffs: Prentice Hall.

Valk, P. E, Bailey, D. L., Townsend, D. W. & Maisey, M. N. (2004). Positron Emission Tomography. Basic Science and Clinical Practice. Berlin: Springer.

Vapnik, V. (1998). Statistical Learning Theory. San Francisco: John Wiley.

Zschocke, S. & Speckmann, E. J. (1993). Basic Mechanisms of the EEG. Boston: Birkhaeuser.

KEY TERMS

Computer Aided Diagnosis (CAD): The use of computers for the interpretation of medical images. Automatic segmentation is one of the crucial task of any CAD product.

Edge Detection: Image processing technique for enhancing the points of an image at which the luminous intensity changes sharply.

Electroencephalography (EEG): Non-invasive diagnostic tool which records the cerebral electrical activity by means of surface electrodes placed on the skull.

Ill-Posedness: Mathematical pathology of differential or integral problems, whereby the solution of the problem does not exist for all data, or is not unique or does not depend continuously on the data. In computation, the numerical effects of ill-posedness are reduced by means of regularization methods.

Image Integration: In medical imaging, combination of different images of the same patient acquired with different modalities and/or according to different geometries.

Magnetic Resonance Imaging (MRI): Imaging modality based on the principles of nuclear magnetic resonance (NMR), a spectroscopic technique used to obtain microscopic chemical and physical information about molecules. MRI can be applied in both functional and anatomical settings.

Magnetoencephalography (MEG): Non-invasive diagnostic tool which records the cerebral magnetic activity by means of superconducting sensors placed on a helmet surrounding the brain.

Segmentation: Image processing technique for distinguishing the different homogeneous regions in an image.

Statistical Learning: Mathematical framework which utilizes functional analysis and optimazion tools for studying the problem of inference.

Tomography: Imaging technique providing two-dimensional views of an object. The method is used in many disciplines and may utilize input radiation of different nature and wavelength. There exist X-ray, optical, microwave, diffraction and electrical impedance tomographies.

Computer Morphogenesis in Self–Organizing Structures

Enrique Fernández-Blanco
University of A Coruña, Spain

Julián Dorado de la Calle
University of A Coruña, Spain

INTRODUCTION

Applying biological concepts to create new models in the computational field is not a revolutionary idea: science has already been the basis for the famous artificial neuron models, the genetic algorithms, etc. The cells of a biological organism are able to compose very complex structures from a unique cell, the zygote, with no need for centralized control (Watson J.D. & Crick F. H. 1953). The cells can perform such process thanks to the existence of a general plan, encoded in the DNA for the development and functioning of the system. Another interesting characteristic of natural cells is that they form systems that are tolerant to partial failures: small errors do not induce a global collapse of the system. Finally, the tissues that are composed by biological cells present parallel information processing for the coordination of tissue functioning in each and every cell that composes this tissue.

All the above characteristics are very interesting from a computational viewpoint. This paper presents the development of a model that tries to emulate the biological cells and to take advantage of some of their characteristics by trying to adapt them to artificial cells. The model is based on a set of techniques known as *Artificial Embryology* (Stanley K. & Miikkulainen R. 2003) or Embryology Computation (Kumar S. & Bentley P.J 2003).

BACKGROUND

The Evoluationary Computation (EC) field has given rise to a set of models that are grouped under the name of Artificial Embryology (AE), first introduced by Stanley and Miikkulainnen (Stanley K. & Miikkulainen R. 2003). This group refers to all the models that try to apply certain characteristics of biological embryonic cells to computer problem solving, i.c. self-organisation, failure tolerance, and parallel information processing.

The work on AE has two points of view. On the one hand can be found the grammatical models based on L-systems (Lindenmayer A. 1968) which do a top-down approach to the problem. On the other hand can be found the chemical models based on the Turing's ideas (Turing A. 1952) which do a down-top approach.

On the last one, the starting point of this field can be found in the modelling of gene regulatory networks, performed by Kauffmann in 1969 (Kauffman S.A. 1969). After that, several works were carried out on subjects such as the complex behaviour generated by the fact that the differential expression of certain genes has a cascade influence on the expressions of others (Mjolsness E., Sharp D.H., & Reinitz J. 1995).

The work performed by the scientific community can be divided into two main branches. The more theoretical branch uses the emulation of cell capabilities such as cellular differentiation and metabolism (Kitano H. 1994; Kaneko K. 2006) to create a model that functions as a natural cell. The purpose of this work is to do an in-depth study of the biological model.

The more practical branch mainly focuses on the development of a cell inspired-model that might be applicable to other problems (Bentley, P.J., Kumar, S. 1999; Kumar, S. 2004). According to this model, every cell would not only have genetic information that encodes the general performance of the system, it would also act as a processor that communicates with the other cells. This model is mainly applied to the solution of simple 3D spatial problems, robot control, generative encoding for the construction of artificial organisms in simulated physical environments and real robots, or to the development of the evolutionary design of hardware and circuits (Endo K., Maeno T. & Kitano H 2003; Tufte G. & Haddow P. C. 2005).

Considering the gene regulatory networks works, the most relevant models are the following: the Kumar and Bentley model (Kumar S. & Bentley P.J 2003), which uses the Bentley's theory of fractal proteins (Bentley, P.J. 1999); for the calculation of protein concentration; the Eggenberger model (Eggenberger P. 1996), which uses the concepts of cellular differentiation and cellular movement to determine cell connections; and the work of Dellaert and Beer (Dellaert F. & Beer R.D. 1996), who propose a model that incorporates the idea of biological operons to control the model expression, where the function assumes the mathematical meaning of a Boolean function.

All these models can be regarded as special cellular automata. In cellular automata, a starting cell set in a certain state will turn into a different set of cells in different states when the same transition function (Conway J.H. 1971) is applied to all the cells during a determined lapse of time in order to control the message concurrence among them. The best known example of cellular automats is Conway's "Game of Life", where this behaviour can be observed perfectly. Whereas the classical conception specifies the behaviour rules, the evolutionary models establish the rules by searching for a specific behaviour. This difference comes from the mathematical origin of the cellular automats, whereas the here presented models are based on biology and embryology.

These models should not be confused with other concepts that might seem similar, such as Gene Expression Programming (GEP) (Ferreira C. 2006). Although GEP codifies the solution in a string, similarly as how it is done in the present work, the solution program is developed in a tree shape, as in classical genetic programming (Koza, J. et. al.1999) which has little or nothing in common with the presented models.

promoter, which identifies the proteins that are needed for gene transcription.

Another remarkable aspect of biological genes is the difference between constitutive genes and regulating genes. The latter are transcribed only when the proteins identified in the promoter part are present. The constitutive genes are always transcribed, unless inhibited by the presence of the proteins identified in the promoter part, acting then as gene oppressors.

The present work has tried to partially model this structure with the aim of fitting some of its abilities into a computational model; in this way, the system would have a structure similar that is similar to the above and will be detailed in the next section.

Proposed Model

Various model variants were developed on the basis of biological concepts. The proposed artificial cellular system is based on the interaction of artificial cells by means of messages that are called proteins. These cells can divide themselves, die, or generate proteins that will act as messages for themselves as well as for neighbour cells.

The system is supposed to express a global behaviour towards the generation of structures in 2D. Such behaviour would emerge from the information encoded in a set of variables of the cell that, in analogy with the biological cells, will be named genes.

One promising application, in which we are working, could be the compact encoding of adaptive shapes, similar to the functioning of fractal growth or the fractal image compression.

ARTIFICIAL EMBRYOGENY MODEL

The cells of a biological system are mainly determined by the DNA strand, the genes, and the proteins contained by the cytoplasm. The DNA is the structure that holds the gene-encoded information that is needed for the development of the system. The genes are activated or transcribed thanks to the protein shaped-information that exists in the cytoplasm, and consist of two main parts: the sequence, which identifies the protein that will be generated if the gene is transcribed, and the

Figure 1. Structure of a system gene

C

The central element of our model is the artificial cell. Every cell has a binary string-encoded information for the regulation of its functioning. Following the biological analogy, this string will be called DNA. The cell also has a structure for the storage and management of the proteins generated by the own cell and those received from neighbourhood cells; following the biological model, this structure is called cytoplasm.

The DNA of the artificial cell consists of functional units that are called genes. Each gene encodes a protein or message (produced by the gene). The structure of a gene has four parts (see **Figure 1**):

• **Sequence:** the binary string that corresponds to the protein that encodes the gene
• **Promoters:** is the gene area that indicates the proteins that are needed for the gene's transcription.
• **Constituent:** this bit identifies if the gene is constituent or regulating
• **Activation percentage (binary value):** the percentage of minimal concentration of promoters proteins inside the cell that causes the transcription of the gene.

The other fundamental element for keeping and managing the proteins that are received or produced by the artificial cell is the cytoplasm. The stored proteins have a certain life time before they are erased. The cytoplasm checks which and how many proteins are needed for the cell to activate the DNA genes, and as such responds to all the cellular requirements for the concentration of a given type of protein. The cytoplasm also extracts the proteins from the structure in case they are needed for a gene transcription.

Model Functioning

The functioning of genes is determined by their type, which can be constituent or regulating. The transcription of the encoded protein occurs when the promoters of the non-constituent genes appear in a certain rate at the cellular cytoplasm. On the other hand, the constituent genes are expressed during all the "cycles" until such expression is inhibited by the present rate of the promoter genes.

Protein Concentration Percent>=
(Distance+1 * Activation Percent   (1)

The activation of the regulating genes or the inhibition of the constituent genes is achieved if the condition expressed by **Eq.1** is fulfilled, where *Protein Concentration Percentage* represents the cytoplasm concentration of the protein that is being considered; *Distance* stands for the Hamming distance between one promoter and the considered protein; and *Activation Percentage* is the minimal percentage needed for the gene activation that is encoded in the gene. This equation is tested on each promoter and each protein. If the condition is fulfilled for all the promoters, that gene is transcribed. According to this, if gene-like promoters exist in a concentration higher than the encoded concentration, they can also induce its transcription, similarly to what happens in biology and therefore providing the model with higher flexibility. If the condition is fulfilled for each promoter, the gene is activated and therefore transcribed.

After the activation of one of the genes, three things can happen: the generated protein may be stored in the cell cytoplasm, it may be communicated to the neighbour cells, or it may induce cellular division (mitosis) and/or death (apoptosis). The different events of a tissue are managed in the cellular model by means of "cellular cycles". Such "cycles" will contain all the actions that can be carried out by the cells, restricting sometimes their occurrence. The "cellular cycles" can be described as follows:

• Actualisation of the life time of proteins in the cytoplasm
• Verification of the life status of the cell (cellular death)
• Calculation of the genes that react and perform the special behaviour that may be associated to them
• Communication between proteins

Solution Search

A classical approach of EC proposes the use of Genetic Algorithms (GA) (Fogel L.J., Owens A. J. & Walsh M.A. 1966; Goldberg D.E. 1989; Holland J.H. 1975) for the optimisation, in this case, of the values of the DNA genes (binary strands). Each individual of the GA population will represent a possible DNA strand for problem solving.

In order to calculate the fitness value for every individual in the GA or the DNA, the strand is introduced into an initial cell or zygote. After simulating during a

Figure 2. (Above) Three promoters and a PCS structure (Below); Example of GA genes association for the encoding of cellular genes
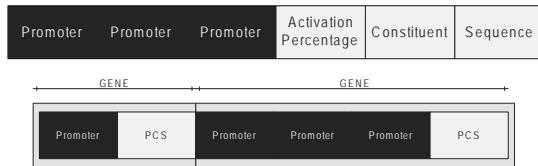
certain number of cycles, the contained information is expressed and the characteristics of the resulting tissue are evaluated by means of various criteria, according to the goal that is to be achieved.

The encoding of the individual genes follows a structure that is similar to the one described in **Figure 2 (Above)**, where the number of promoters of each gene may, vary but the white and indivisible section "Activation Percentage – Constituent – Sequence" (PCS) must always be present. The PCS sections determine the genes of the individual, and the promoter sections are associated to the PCS sections, as shown in **Figure 2(Below)**.

The search of a set of structures similar to those shown in **Figure 2** required the adaptation of the crossover and mutation GA operations to this specific problem. Since the length of the individuals is variable, the crossover had to be performed according to these lengths. When an individual is selected, a random percentage is generated to determine the crossover point of that individual. After selecting the section in that position, a crossover point is chosen for the section selected in the other parent. Once this has been done, the crossover point selection process is repeated in the second selected parent in the same position as in the previous individual. From this stage on, the descendants are composed in the traditional way, since they are two strings of bits. We could execute a normal bit strings crossover, but the previously mentioned steps guarantee that the descendants are valid solutions for the DNA strands transformation.

With regards to mutation, it should be mentioned that the types of the promoter or PCS sections are identified according to the value of the first string bit. Bearing that in mind, together with the variable length of individuals, the mutation operation had to be adapted

so that it could modify not only the number of these sections, but also the value of a given section.

The probability of executing the mutation is usually low, but this time it even had to be divided into the three possible mutation operations that the system contemplates. Various tests proved that the most suitable values for the distribution of the different mutation operations, after the selection of a position for mutation, were the following: for 20% of the opportunities, a section (either a promoter or a PCS) is added; for another 20%, the existing section is removed; and finally, for the remaining 60% of the opportunities, the value of one of the bits of the section is randomly changed. The latter may provoke not only the change of one of the values, but also the change of the section type: if the bit that identifies the section type is changed, the information of that section varies. For instance, if a promoter section turns into a PCS section, the promoter sequence turns into the gene sequence, and constitutive and activation percentage values are generated.

After reaching this development level and presenting the test set in (Fernández-Blanco E., Dorado J., Rabuñal J.R., Gestal M. & Pedreira N. 2007), the authors concluded that the bottleneck of the model turned out to be the development of the evaluation functions, since in every new figure the development of the function was time-consuming and not reusable.

In order to solve this problem, the evaluation function was developed according to the concept of a correction template. From the tissue that is developed by the DNA that is being evaluated, the centroid is calculated. This point would be the center of the solution template, which is merely a matrix of Boolean values representing the figure that is aimed at. The template could be (and usually is) smaller than the development environment of the tissue, which means that every cell that may not be covered by the template will contribute to the tissue error with 1.0. The remaining tissue, covered by the

Figure 3. Tissue + Template. Example of Template use.

C
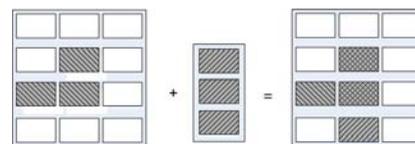
template, will execute the NEXOR Boolean operation in order to obtain the number of differences between the template and the tissue. Each difference contributes with a value of 1.0. to tissue error.

Figure 3 illustrates the use of this method. We can observe that the error of this tissue with regard to the template is 2, since we generated a cell that is not contemplated by the template, whereas another cell that is present in the template is really missing.

FUTURE TRENDS

The model could also include new characteristics such as the displacement of cells around their environment, or a specialisation operator that blocks pieces of DNA during the expression of its descendants, as happens in the natural model.

Finally, this group is currently working in one of the possible applications of this model: its use for image compression similarly as fractal compression works. The fractal compression searches the parameters of a fractal formula that encodes itself the starting image. The present model searches the gene sequence that might result in the starting image. In this way, the method based on template that has been presented in this paper can be used for performing that search, using the starting image as template.

CONCLUSION

Taking into account the here developed model, we can say that the use of certain properties of biological cellular systems is feasible for the creation of artificial structures that might be used in order to solve certain computational problems.

Some behaviours of the biological model have been also observed in the artificial model: information redundancy in DNA, stability after achieving the desired shape, or variability in gene behaviour.

REFERENCES

Bentley, P.J., Kumar, S. (1999) *Three ways to grow designs: A comparation of three embryogenies for an evolutionary design problem*. In Proceedings of Genetic and Evolutionay Computation.

Conway J.H. (1971) *Regular Algebra and Finite Machines*. Chapman and Hall, Ltd., London

Dellaert F. & Beer R.D. (1996) *A Developmental Model for the Evolution of Complete Autonomous Agent* In From animals to animats: Proceedings of the Forth International Conference on Simulation of Adaptive Behavior, Massachusetts, September 9-13, pp. 394-401, MIT Press.

Eggenberger P. (1996) *Cell Interactions as a Control Tool of Developmental Processes for Evolutionary Robotics*. In From animals to animats: Proceedings of the Forth International Conference on Simulation of Adaptive Behavior, Massachusetts, September 9-13, pp. 440-448, MIT Press.

Endo K., Maeno T. & Kitano H. (2003): *Co-evolution of morphology and walking pattern of biped humanoid robot using evolutionary computation -designing the real robot*. ICRA 2003: 1362-1367

Fernández-Blanco E., Dorado J., Rabuñal J.R., Gestal M. & Pedreira N. (2007) *A New Evolutionary Computation Technique for 2D Morphogenesis and Information Processing*. WSEAS Transactions on Information Science & Applications vol. 4(3) pp.600-607, WSEAS Press.

Ferreira C. (2006) *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence* Springer, Berlin.

Fogel, L.J., Owens, A. J. & Walsh, M.A. (1966) *Artificial Intelligence through Simulated Evolution*. Wiley, New York.

Goldberg, D.E. (1989). *Genetics Algorithms in Search*. Optimization and Machine Learning. Addison-Wesley.

Holland, J.H. (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MA, USA.

Kaneko K. (2006) *Life: An Introduction to Complex Systems Biology*. Springer Complexity: Understanding Complex Systems, Springer Press.

Kauffman, S.A. (1969) *Metabolic stability and epigenesis in randomly constructed genetic nets*. Journal of Theoretical Biology 22 pp. 437-467.

Kitano, H. (1994). *Evolution of Metabolism for Morphogenesis*. In Artificial Life IV, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, edited by Brooks, R. and Maes, P. MIT Press, Cambridge,MA.

Koza, J. et. al.(1999). *Genetic Programming III: Darwin Invention and Problem Solving*. MIT Press, Cambridge, MA.

Kumar, S. & Bentley P.J. (editors) (2003). *On Growth, Form and Computers*. Academic Press. London UK.

Kumar, S. (2004). *Investigating Computational Models of Development for the Construction of Shape and Form*. PhD Thesis. Department of Computer Science, University Collage London.

Lindenmayer, A. (1968) *Mathematical models for cellular interaction in development: Part I and II*. Journal of Theorical Biology. Vol. 18 pp. 280-299, pp. 300-315.

Mjolsness, E., Sharp, D.H., & Reinitz, J. (1995) *A Connectionist Model of Development*. Journal of Theoretical Biology 176: 291-300.

Stanley, K. & Miikkulainen, R. (2003) *A Taxonomy for Artificial Embryogeny*. In Proceedings Artificial Life 9, pp. 93-130. MIT Press.

Tufte, G. & Haddow, P. C. (2005) *Towards Development on a Silicon-based Cellular Computing Machine*. Natural Computing 4 vol. 4: pp.387-416.

Turing, A.(1952) *The chemical basis of morphogenesis*. Philosofical Transactions of the Royal Society B, vol.237, pp. 37-72

Watson J.D. & Crick. F. H. (1953) *Molecular structure of Nucleic Acids*. Nature vol. 171, pp. 737–738.

KEY TERMS

Artificial Cell: Each of the elements that process the orders codified into the DNA.

Artificial Embryogeny: The term overlaps all the processing models which use biological development ideas as inspiration for its functioning.

Cellular Cycle: Cellular development time unit which limits the ocurrents number of certain cellular development actions.

Cytoplasm: Part of an artificial cell which is responsible of management the protein-shaped messages.

DNA: Set of rules which are responsible of the cell behaviour.

Gene: Each of the rules which codifies one action of the cell.

Protein: This term identifies every kind of the messages that receives an artificial cell.

Zygote: The initial cell from where a tissue is generated using the DNA information.

Computer Vision for Wave Flume Experiments

C

Óscar Ibáñez
University of A Coruña, Spain

Juan Ramón Rabuñal Dopico
University of A Coruña, Spain

INTRODUCTION

During the past several decades, a number of attempts have been made to contain oil slicks (or any surface contaminants) in the open sea by means of a floating barrier. Many of those attempts were not very successful especially in the presence of waves and currents. The relative capabilities of these booms have not been properly quantified for lack of standard analysis or testing procedure (Hudon, 1992). In this regard, more analysis and experimental programs to identify important boom effectiveness parameters are needed.

To achieve the desirable performance of floating booms in the open sea, it is necessary to investigate the static and dynamic responses of individual boom sections under the action of waves; this kind of test is usually carried out in a wave flume, where open sea conditions can be reproduced at a scale.

Traditional methods use capacitance or conductivity gauges (Hughes, 1993) to measure the waves. One of these gauges only provides the measurement at one point; further, it isn't able to detect the interphase between two or more fluids, such as water and a hydrocarbon. An additional drawback of conventional wave gauges is their cost.

Other experiments such as velocity measurements, sand concentration measurements, bed level measurements, breakwater's behaviour, etc… and the set of traditional methods or instruments used in those experiments which goes from EMF, ADV for velocity measurements to pressure sensors, capacity wires, acoustic sensors, echo soundings for measuring wave height and sand concentration, are common used in wave flume experiments. All instruments have an associate error (Van Rijn, Grasmeijer & Ruessink, 2000), and an associate cost (most of them are too expensive for a lot of laboratories that can not afford pay those amount of money), certain limitations and some of them need a large term of calibration.

This paper presents another possibility for wave flume experiments, computer vision, which used a cheap and affordable technology (common video cameras and pc's), it is calibrated automatically (once we have developed the calibration task), is a non-intrusive technology and its potential uses could takes up all kind experiments developed in wave flumes. Are artificial vision's programmers who can give computer vision systems all possibilities inside the visual field of a video camera. Most experiments conducted in wave flumes and new ones can be carried out programming computer vision systems. In fact, in this paper, a new kind of wave flume experiment is presented, a kind of experiment that without artificial vision technology it couldn't be done.

BACKGROUND

Wave flume experiments are highly sensitive to whatever perturbation; therefore, the use of non-invasive measurement methodologies is mandatory if meaningful measures are desired. In fact, theoretical and experimental efforts whose results have been proposed in the literature have been mainly conducted focusing on the equilibrium conditions of the system (Niederoda and Dalton, 1982), (Kawata and Tsuchiya, 1988).

In contrast with most traditional methods used in wave flume experiments computer vision systems are non-invasive ones since the camera is situated outside the tank and in addition provide better accuracy than most traditional instruments.

The present work is part of a European Commission research project, "Advanced tools to protect the Galician and Northern Portuguese coast against oil spills at sea", in which a number of measurements in a wave flume must be conducted, such as the instantaneous position of the water surface or the motions (Milgran, 1971) of a floating containment boom to achieve these

objectives, a non-intrusive method is necessary (due to the presence of objects inside the tank) and the method has to be able to differentiate between at least two different fluids, with the oil slick in view.

Others works using image analysis to measure surface wave profile, have been developed over the past ten years (e.g., Erikson and Hanson, 2005; García, Herranz, Negro, Varela & Flores, 2003; Javidi and Psaltis, 1999; Bonmarin, Rochefort & Bourguel, 1989; Zhang, 1996), but they were developed neither with a real-time approach nor as non-intrusive methods. In some of these techniques it is necessary to colour the water with a fluorescent dye (Erikson and Hanson, 2005), which is not convenient in most cases, and especially when two fluids must be used (Flores, Andreatta, Llona & Saavedra, 1998).

A FRAMEWORK FOR MEASURING WAVES LEVEL IN A WAVE FLUME WITH ARTIFICIAL VISION TECHNIQUES

Following is presented an artificial vision system (Ibáñez, Rabuñal, Castro, Dorado, Iglesias & Pazos, 2007) which obtains the free surface position in all points of the image, from which the wave heights can be computed. For this aim we have to record a wave tank (see laboratory set-up in section 2) while it is generating waves and currents (a scale work frame), and after that we have to use the frames which make up the image to obtain the crest of the water (using computer vision techniques described in section 3) and translate the distances in the image to real distances (taking into account image rectification, see section 1).

Image Rectification

Lens distortion is an optical error in the lens that causes differences in magnification of the object at different points on the image; straight lines in the real world may appear curved on the image plane (Tsai, 1987). Since each lens element is radially symmetric, and the elements are typically placed with high precision on the same optical axis, this distortion is almost always radially symmetric and is referred to as radial lens distortion (Ojanen, 1999). There are two kinds of lens distortion: barrel distortion and pincushion distortion. Most lenses exhibit both properties at different scales.

Figure 1. Template to image rectification. Crosses are equidistant with a 4cm separation.
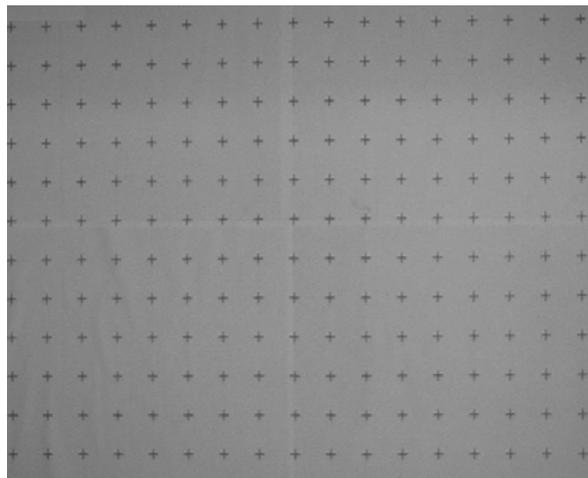
To avoid lens distortion error and to provide a tool for transforming image distances (number of pixels) to real distances (mm) it is necessary to follow a rectification procedure.

Most image rectification procedures involve a two step process (Ojanen, 1991). (Holland, Holman & Sallenger, 1991): calibration of intrinsic camera parameters, and correction for a camera's extrinsic parameters (i.e., the location and rotation in space).

However, in our case we are only interested in transforming pixel measurements into real distances (mm). Transforming points from a real world surface to a non-coplanar image plane would imply an operator which, when applied to all frames, would considerably slow down the total process, which is not appropriate for our real-time approach.

So a .NET routine was developed to create a map with the corresponding factor (between pixel and real distances) for each group of pixels (four nearest control points on the target). Inputs to the model are a photographed image of the target sheet (see fig.1), and target dimensions (spacing between control points in the x- and y-directions).

Laboratory Set-Up and Procedure

The experiment was conduced in a 17.29-m long wave flume at the Centre of Technological Innovation in

Figure 2. Laboratory set-up diagram
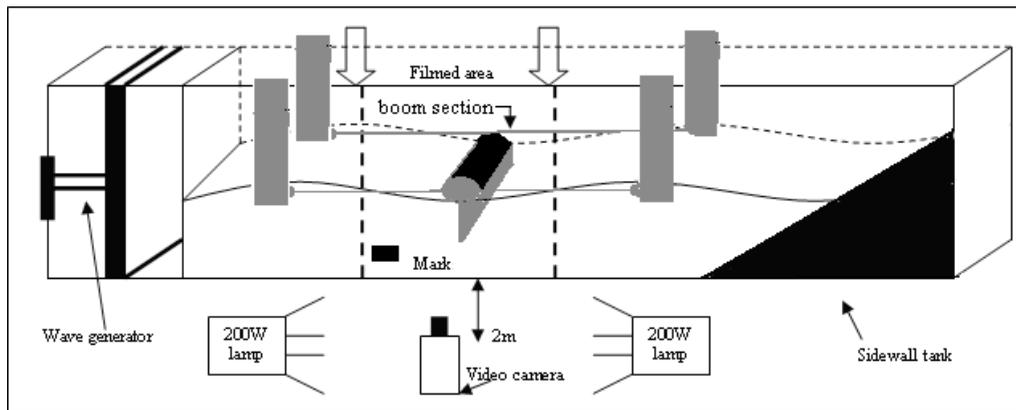
Construction and Civil Engineering (CITEEC), in the University of A Coruña, Spain. The flume section is 77 cm (height) x 59.6 cm (width). Wave generation is conducted by means of a piston-type paddle. A wave absorber is located near its end wall to prevent wave reflection. It consists of a perforated plate with a length of 3.04 m, which can be placed at different slopes. The experimental set-up is shown in fig. 2.

With the aim of validating the system, solitary waves were generated and measured on the base of images recorded by a video camera mounted laterally, which captured a flume length of 1 m. The waves were also measured with one conductivity wave gauge located within the flume area recorded by the video camera. These gauges provide an accuracy of ± 1 mm at a maximum sampling frequency of 30 Hz.

A video camera, Sony DCR-HC35E, was used in turn to record the waves; it worked on the PAL Western Europe standard, with a resolution of 720 x 576 pixels, recording 25 frames per second.

The camera was mounted on a standard tripod and positioned approximately 2 m from the sidewall of the tank (see fig. 2). It remained fixed throughout the duration of a test. The procedure is as follows:

- Place one mark on the glass sidewall of the flume, on the bottom of the filmed area (see fig. 2);
- Place a template with equidistant marks (crosses) in a vertical plane parallel to the flume sidewall (see fig 1).
- Position the camera at a distance from the target plane (i.e., tank sidewall) depending on desired resolution.

- Adjust the camera taking into account the template's marks.
- Provide uniform and frontal lighting for the template.
- Film the template.
- Provide uniform lighting on the target plane and a uniformly colored background on the opposite sidewall (to block any unwanted objects from the field of view);
- Start filming.

The mark was placed horizontally on the glass sidewall of the flume, on the bottom of the filmed area in order to know a real distance between the bed of the tank and this mark, to avoid filming the bed of the tank and thus to film a smaller area (leading to a better resolution).

With regard to the lighting of the laboratory it is necessary to avoid direct lighting and consequently we can work without gleam and glints.

To achieve this kind of lighting, all lights in the laboratory were turned off and two halogen lamps of 200W were placed on both sides of the filmed area, one in front the other (see fig. 2).

Video Image Post-Processing

Image capture was carried out on a PC, Pentium 4, 3.00 GHz and 1.00 GB de RAM memory with the Windows XP platform. Filmed with the Sony DCR-HC35E[1], a high-speed interface card, IEEE 1394 FireWireTM, was used to transfer digital data from the camcorder to the computer, and the still images were kept in the

uncompressed bitmap format so that information would not be lost. De-interlacing was not necessary because of the quality of the obtained images and everything was done on a real-time approach.

An automatic tool for measuring waves from consecutive images was developed. The tool was developed under .NET framework, using C++ language and OpenCV (Open Source Computer Vision Library, developed by Intel[2]) library. The computer vision procedure is as follows:

- Extract a frame from the video.
- Using different computer vision algorithms get the constant "pixel to mm" for each pixel.
- Using different computer vision algorithms, the crest of the wave is obtained.
- Work out the corresponding height, for all the pixels in the crest of the wave.
- Supply results.
- Repeat the process until the video finish.

With regard to get the constant "pixel to mm", a template with equidistant marks (crosses) is placed right up the glass sidewall of the tank and is filmed. Then a C++ routine recognize de centre of the crosses.

Results

A comparison of data extracted from video images with data measured by conventional instruments was done. The comparisons are not necessarily meant to validate the procedure as there are inherent errors with conventional instruments, as well; rather, the comparisons aim to justify the use of video images as an alternative method for measuring wave and profile change data.

Different isolated measurements with conductivity gauge were done at the same time the video camera was recording. Then results from both methods were compared.

The process followed to measure with conductivity gauge and the artificial vision system at the same time involves recognizing one point in x-axis (in the record video) where the gauge is situated (one color mark was pasted around the gauge to make easier this task) and after knowing the measure point of the gauge we create a file with the height of the wave in this x point for each image in the video. While the video-camera is recording one file with gauge measure is created. Once we have both measure files we have two determine manually the same time point in both files (due to the difficulty to initialize both systems at the same time). Now, we can compare both measurements.

A lot of tests were done with different wave parameters for wave period, wave height and using regular (sine form) and irregular waves. Test with waves between 40mm and 200mm of height were done.

Using the camera DCR-HC35E, figure 3 shows one example of a test done, where the used wave was an irregular one, with a maximum period of 1s and a

Figure 3. Temporal sequence of measurement by sensor and image analysis

C
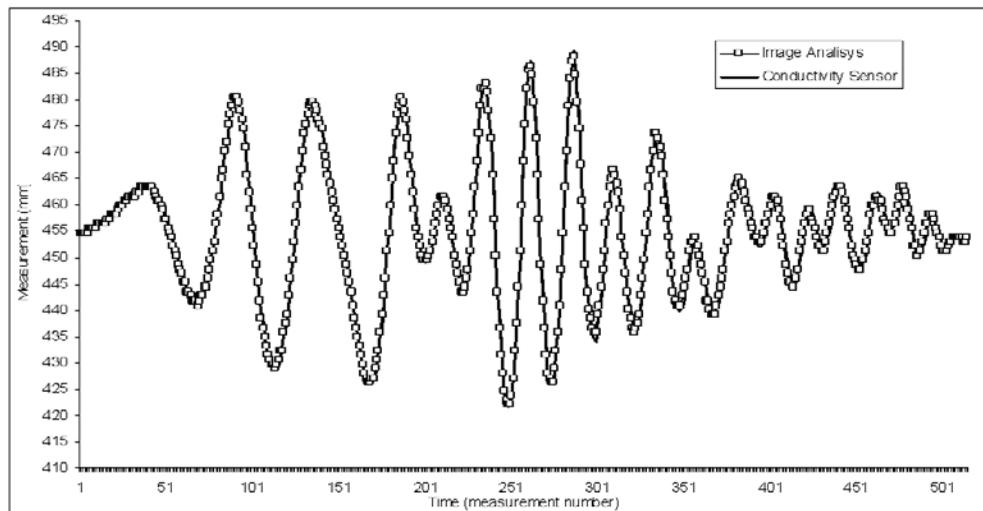
maximum value for wave amplitude of 70mm, excellent results were obtained as it can be seen in figure 4, where both measurements (conductivity sensor and video analysis) are quite similar.

The correlation between sensor and video image analysis measurements has an associated mean square error of 0.9948.

In spite of these sources of error, after several tests, the average error between conductivity sensor measurements and video analysis is 0.8 mm with camera DCR-HC35E, a lot better compared with the 5 mm average error obtained in the best work done until this moment (Erikson and Hanson, 2005). But it isn't an indicative error because of the commented source of errors taken into account in this study, however the estimated real error from this video analysis system is 1 mm, that is to say, the equivalence between one pixel and a real distance, and in our case (with the commented video camera and distance from the tank) one pixel is equivalent to nearly 1mm. This error could be improvable with a camera which allows a better resolution or focusing a smaller area.

FUTURE TRENDS

This is the first part of a bigger system which is capable of measuring the motions of a containment boom section in the vertical axis and its slope angle (Kim, Muralidharan, Kee, Jonson, & Seymour, 1998). Furthermore the system would be capable of making a distinction between the water and a contaminant, and thus would identify the area occupied by each fluid.

Another challenge is to test this system in other work spaces with different light conditions (i.e., in a different wave flume).

CONCLUSION

An artificial vision system was developed for these targets because these systems are non-intrusive and can separate a lot of different objects or fluids (anything that a human eye can differentiate) in the image and a non-intrusive method is necessary.

Other interesting aspects that these systems provide are:

- Cheaper price than traditional systems of measurement.
- Easier and faster to calibrate.
- It is unnecessary to mount an infrastructure to know what happens at different points of the tank (only one camera instead of an array of sensors).
- As the system is a non-intrusive one, it doesn't distort the experiments and their measurements.
- Provide high accuracy.
- Finally, this system is an innovation idea of applying computer vision techniques to civil engineering area and specifically in ports and coasts field. No similar works have been developed.

REFERENCES

Baglio, S. & Foti, E., 2003. *Non-invasive measurements to analyze sandy bed evolution under sea waves action*. Instrumentation and Measurement, IEEE Transactions on. Vol. 52, Issue: 3, pp. 762-770.

Bonmarin P., Rochefort R. & Bourguel M., 1989. *Surface wave profile measurement by image analysis*, Experiments in Fluids, Vol.7, No.1, Springer Berlin / Heidelberg, pp. 17-24.

Erikson, L. H. & Hanson, H., 2005. *A method to extract wave tank data using video imagery and its comparison to conventional data collection technique*, Computers & Geosciences, Vol.31, pp. 371-384.

Flores, H., Andreatta, A., Llona & G., Saavedra, I., 1998. *Measurements of oil spill spreading in a wave tank using digital image processing*, The 1998 1st International Conference on Oil and Hydrocarbon Spills, Modelling, Analysis and Control, Oil Spill, pp. 165-173.

García, J., Herranz, D., Negro, V., Varela & O., Flores, J., 2003. *Tratamiento por color y video*. VII Jornadas Españolas de Costas y Puertos.

Haralick, R. & Shapiro L., 1992. *Computer and Robot Vision*.Vol 1, Addison-Wesley Publishing Company, Chap 5, pp 174-185.

Harris, C. & Stephens, M. J., 1988. *A combined corner and edge detector*. In Alvey Vision Conference, pages 147–152.

Holland, K.T., Holman, R.A. & Sallenger, A.H., 1991. *Estimation of overwash bore velocities using video techniques*. In: Proceedings of the Third International Symposium on Coastal Engineering and the Science of Coastal Sediment Processes, Seattle, WA, pp. 489–496.

Hu, M. K., 1962. *Visual pattern recognition by moment invariants*, IRE Trans. Inform. Theory, vol. 8, pp. 179–187.

Hudon, T., 1992. *Wave tank testing of small scale boom sections*. Private communication.

Hughes, 1993. *Laboratory wave reflection analysis using co-located gages*. Coastal Engineering. Vol. 20, no. 3-4, pp. 223-247.

Ibáñez O., Rabuñal J., Castro A., Dorado J., Iglesias G, Pazos A., 2007. *A framework for measuring waves level in a wave tank with artificial vision techniques*. WSEAS Transactions on Signal Processing Editorial: WSEAS Press Volume 3, Issue 1, pp 17-24.

Javidi, B. & Psaltis, D., 1999. *Image sequence analysis of water surface waves in a hydraulic wind wave tank*, Proceedings of SPIE, Vol.3804, Psaltis, pp. 148-158.

Kawata Y. & Tsuchiya Y., 1988. *Local scour around cylindrical piles due to waves and currents*. Proc. 21st Coast. Eng. Conf., Vol. 2, ASCE, New York, pp. 1310–1322.

Kim, M. H., Muralidharan, S., Kee, S. T., Jonson, R. P. & Seymour, R. J., 1998. *Seakeeping performance of a containment boom section in random waves and currents*. Ocean Engng, Vol 25, Nos. 2-3, pp. 143-172.

Konstantinos N. Plataniotis & Anastasios N Venetsanopoulos, 2000. *Color Image Processing and Applications*. Springer.

Milgran, J. H., 1971. *Forces and motions of a flexible floating barrier*. Journal of Hydronautics 5, 41-51.

Mukundan, R. & Ramakrishman K. R., 1998. *Moment Functions in Image Analysis: Theory and Application*, Singapore: World Scientific.

Niederoda, A. W., & Dalton C., 1982. *A review of the fluid mechanics of ocean scour*. Ocean Eng., Vol. 9, pp. 159-170.

Ojanen, H., 1999. *Automatic correction of lens distortion by using digital image processing*. http://www.iki.fi

Schmid, C., Mohr, R. & Bauckhage., 2000. C. *Evaluation of interest point detectors*. International Journal of Computer Vision, 37(2):151-172

Tsai, R.Y., 1987. *A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-theshelf TV cameras and lenses*. Journal of Robotics and Automation RA (3), 323–344.

Van Rijn, Grasmeijer & Ruessink, 2000. *Measurement errors of instruments for velocity, wave height, sand concentration and bed levels in field conditions*. Coast 3D, November.

Vernon D., 1991. *Machine Vision*. Prentice-Hall, pp 78-79.

Zhang, X., 1996. *An algorithm for calculating water surface elevations from surface gradient image data*, Experiments in Fluids, Vol.21, No.1, Springer Berlin / Heidelberg, pp. 43-48.

KEY TERMS

Color Spaces: (Konstantinos & Anastasios, 2000) supply a method to specify, sort and handle colors. These representations match n-dimensional sorts of the color feelings (n-components vector). Colors are represented by means of points in these spaces. There are lots of colors spaces and all of them start from the same concept, the Tri-chromatic theory of primary colors, red, green and blue.

Dilation: The dilation of an image by a structuring element 'Y' is defined as the maximum value of all the pixels situated under the structuring element

$$\varepsilon_Y(f)(x,y) = \min_{(s,t)\in Y} f(x+s, y+t).$$

The basic effect of this morphological operator the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels

grow in size while holes within those regions become smaller.

Erosion: The basic effect of the operator on a binary image is to reduce the definition of the objects. The erosion in the point (x,y) is the minimum value of all the points situated under the window, which is defined by the structuring element 'Y' that travels around the image:

$$\varepsilon_Y(f)(x, y) = \max_{(s,t)\in Y} f(x + s, y + t).$$

Harris Corner Detector: A popular interest point detector (Harris and Stephens, 1988) due to its strong invariance to (Schmid, Mohr, & Bauckhage, 2000): rotation, scale, illumination variation and image noise. The Harris corner detector is based on the local auto-correlation function of a signal; where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions.

Image Moments: (Hu, 1963; Mukundan and Ramakrishman, 1998) they are certain particular weighted averages (moments) of the image pixels' intensities, or functions of those moments, usually chosen to have some attractive property or interpretation. They are useful to describe objects after segmentation. Simple properties of the image which are found via image moments include area (or total intensity), its centroid, and information about its orientation.

Morphological Operators: (Haralick and Shapiro, 1992; Vernon, 1991) Mathematical morphology is a set-theoretical approach to multi-dimensional digital signal or image analysis, based on shape. The signals are locally compared with so-called structuring elements of arbitrary shape with a reference point.

Videometrics: (Tsai, 1987) can loosely be defined as the use of imaging technology to perform precise and reliable measurements of the environment.

ENDNOTES

[1] http://www.sony.es/view/ShowProduct.action?product=DCR-C35E&site=odw_es_ES&pageType=Overview&category=CAM+Mini DV

[2] http://www.intel.com/technology/computing/opencv/

Conditional Hazard Estimating Neural Networks

Antonio Eleuteri
Royal Liverpool University Hospital, UK

Azzam Taktak
Royal Liverpool University Hospital, UK

Bertil Damato
Royal Liverpool University Hospital, UK

Angela Douglas
Liverpool Women's Hospital, UK

Sarah Coupland
Royal Liverpool University Hospital, UK

INTRODUCTION

Survival analysis is used when we wish to study the occurrence of some event in a population of subjects and the time until the event of interest. This time is called *survival time* or *failure time*. Survival analysis is often used in industrial life-testing experiments and in clinical follow-up studies. Examples of application include: time until failure of a light bulb, time until occurrence of an anomaly in an electronic circuit, time until relapse of cancer, time until pregnancy.

In the literature we find many different modeling approaches to survival analysis. Conventional parametric models may involve too strict assumptions on the distributions of failure times and on the form of the influence of the system features on the survival time, assumptions which usually extremely simplify the experimental evidence, particularly in the case of medical data (Cox & Oakes, 1984). In contrast, semi-parametric models do not make assumptions on the distributions of failures, but instead make assumptions on how the system features influence the survival time (the usual assumption is the proportionality of hazards); furthermore, these models do not usually allow for direct estimation of survival times. Finally, non-parametric models usually only allow for a qualitative description of the data on the population level.

Neural networks have recently been used for survival analysis; for a survey on the current use of neural networks, and some previous attempts at neural network

survival modeling we refer to (Bakker & Heskes, 1999), (Biganzoli et al., 1998), (Eleuteri et al., 2003), (Lisboa et al., 2003), (Neal, 2001), (Ripley & Ripley, 1998), (Schwarzer et al. 2000).

Neural networks provide efficient parametric estimates of survival functions, and, in principle, the capability to give personalised survival predictions. In a medical context, such information is valuable both to clinicians and patients. It helps clinicians to choose appropriate treatment and plan follow-up efficiently. Patients at high risk could be followed up more frequently than those at lower risk in order to channel valuable resources to those who need them most. For patients, obtaining information about their prognosis is also extremely valuable in terms of planning their lives and providing care for their dependents.

In this article we describe a novel neural network model aimed at solving the survival analysis problem in a continuous time setting; we provide details about the Bayesian approach to modeling, and a sample application on real data is shown.

BACKGROUND

Let T denote an absolutely continuous positive random variable, with distribution function P, representing the time of occurrence of an event. The survival function, $S(t)$, is defined as:

$S(t) = \Pr(T > t)$,

that is, the probability of surviving beyond time t. We shall generally assume that the survival function also depends on a set of covariates, represented by the vector x (which can itself be assumed to be a random variable). An important function related to the survival function is the *hazard rate* (Cox & Oakes, 1984), defined as:

$h_r(t) = P'(t)/S(t)$

where P' is the density associated to P. The hazard rate can be interpreted as the instantaneous force of mortality.

In many survival analysis applications we do not directly observe realisations of the random variable T; therefore we must deal with a missing data problem. The most common form of missingness is *right censoring*, i.e., we observe realisations of the random variable:

$Z = \min(T, C)$,

where C is a random variable whose distribution is usually unknown. We shall use a censoring indicator d to denote whether we have observed an event ($d=1$) or not ($d=0$). It can be shown that inference does not depend on the distribution of C (Cox & Oakes, 1984).

With the above definitions in mind we can now formulate the log-likelihood function necessary for statistical inference. We shall omit the details, and only report the analytical form:

$$L = \sum_i d_i \log h_r(t_i, x_i) - \int_0^{t_i} h_r(u, x_i) \mathrm{d}u.$$

For further details, we refer the reader to (Cox & Oakes, 1984).

CONDITIONAL HAZARD ESTIMATING NEURAL NETWORKS

Neural Network Model

The neural network model we used is the Multi-Layer Perceptron (MLP) (Bishop, 1995):

$$a(t, x; w) = b_0 + \sum_k v_k g(u_k^T x + u_0 t + b_k)$$

where $g()$ is a sigmoid function, and $w = \{b_0, v, u, u_0, b\}$ is the set of network parameters. The MLP output defines an analytical model for the logarithm of the hazard rate function:

$$a(t, x; w) \triangleq \log h_r(t, x)$$

We refer to this continuous time model as Conditional Hazard Estimating Neural Network (CHENN).

Bayesian Learning of the Network Parameters

The Bayesian learning framework offers several advantages over maximum likelihood methods commonly used in neural network learning (Bishop, 1995), (MacKay, 1992), among which the most important are automatic regularization and estimation of error bars on predictions.

In the conventional maximum likelihood approach to training, a single weight vector is found, which minimizes the error function; in contrast, the Bayesian scheme considers a probability distribution over weights w. This is described by a prior distribution $p(w)$ which is modified when we observe a dataset D. This process can be expressed by Bayes' theorem:

$$p(w \mid D) = \frac{p(D \mid w) p(w)}{p(D)}.$$

To evaluate the posterior distribution, we need expressions for the likelihood $p(D|w)$ (which we have already shown) and for the prior $p(w)$.

The prior over weights should reflect the knowledge, if any, we have about the mapping we want to build. In our case, we expect the function to be very smooth, so an appropriate prior might be:

$$p(w) \propto \exp\left(-\frac{1}{2} \sum_k \alpha_k w_k^T w_k \right)$$

which is a multivariate normal density with zero mean and diagonal covariance matrix with elements $1/\alpha_k$. In this way, weights centered on zero have higher probability, a fact which encourages very smooth functions.

Note that the prior is parametric, and the regularization parameters α_k (which are inverse variances) are called *hyperparameters*, because they control the distribution of the network parameters.

Note also that the prior is specialized for different groups of weights by using different regularization parameters for each group; this is done to preserve the scaling properties of network mappings (Bishop, 1995), (MacKay, 1992). This prior is called Automatic Relevance Determination (ARD). This scheme defines a model whose prior over the parameters embodies the concept of relevance, so that the model is effectively able to infer which parameters are relevant based on the training data and then switch the others off (or at least reduce their influence on the overall mapping).

The ARD modeling scheme in the case of the CHENN model defines weight groups for the inputs, the output layer weights, and the biases.

Once the expressions for the prior and the noise model are given, we can evaluate the posterior:

$$p(w \mid D) = \frac{1}{Z} \exp\left(L - \frac{1}{2} \sum_k \alpha_k w_k^T w_k \right).$$

This distribution is usually very complex and multimodal (reflecting the nature of the underlying error function, the term -*L*); and the determination of the normalization factor (also called the *evidence*) is very difficult. Furthermore, the hyperparameters must be integrated out, since they are only used to determine the form of the distributions.

A solution is to integrate out the parameters separately from the hyperparameters, by making a Gaussian approximation; then, searching for the mode with respect to the hyperparameters (Bishop, 1995), (MacKay, 1992). This procedure gives a good estimation of the *probability mass* attached to the posterior, in particular for distributions over high-dimensional spaces, which is the case for large networks.

The Gaussian approximation is in practice derived by finding a maximum of the posterior distribution, and then evaluating the curvature of the distribution around the maximum:

$$w_{MP} = \arg\max_w L - \frac{1}{2} \sum_k \alpha_k w_k^T w_k,$$

$$A = \nabla\nabla_w \left[-L + \frac{1}{2} \sum_k \alpha_k w_k^T w_k \right]_{w=w_{MP}}.$$

The approximation thus is:

$$p(w \mid D) \approx \frac{1}{Z_{MP}} \exp\left(-\frac{1}{2} (w - w_{MP})^T A^{-1} (w - w_{MP}) \right)$$

where the normalisation constant is simply evaluated from usual multivariate normal formulas.

The hyperparameters are calculated by finding the maximum of the approximate *evidence* Z_{MP}. Alternate maximization (by using a nonlinear optimization algorithm) of the posterior and evidence is repeated until a self consistent solution $\{w_{MP}, \alpha_k\}$ is found.

The full Bayesian treatment of inference implies that we do not simply get a pointwise prediction for functions *f(x,t;w)* of a model output, but a full distribution. Such predictive distributions have the form:

$$p(f(x,t \mid D)) = \int f(x,t \mid w) p(w \mid D) dw.$$

The above integrals are in general not analytically tractable, even when the posterior distribution over the parameters is Gaussian. However, it is usually enough to find the moments of the predictive distribution, in particular its mean and variance. A useful approximation is given by the delta method. Let *f(w)* be the function (of *w*) we wish to approximate. By Taylor expanding to first order around w_{MP}, we can write:

$$f(w) \approx f(w_{MP}) + (w - w_{MP})^T \nabla_w f(w)_{w=w_{MP}}.$$

Since this is a linear function of *w*, it will still be normally distributed under the Gaussian posterior, with mean and variance:

$$\mathbf{E}[f(w)] = f(w_{MP})$$

$$\mathbf{Var}[f(w)] = \nabla_w^T f \, A^{-1} \, \nabla f_w.$$

Error bars are simply obtained by taking the square root of the variance. We emphasize that it is important to evaluate first and second order information to understand the overall *quality* and *reliability* of a model's predictions. Error bars also provide hints on the distribution of the patterns (Williams et al., 1995) and can therefore be useful to understand whether a model is extrapolating its predictions. Furthermore, they can offer suggestions for the collection of future data (Williams et al., 1995; MacKay, 1992).

A Case Study: Ocular Melanoma

We show now an application of the CHENN model to the prognosis of all-cause mortality in ocular melanoma.

Intraocular melanoma occurs in a pigmented tissue called the uvea, with more than 90% of tumours involving the choroid, beneath the retina. About 50% of patients die of metastatic disease, which usually involves the liver.

Estimates for survival after treatment of uveal melanoma are mostly derived and reported using Cox analysis and Kaplan-Meier (KM) survival curves (Cox & Oakes). As a semiparametric model, the Cox method, however, usually utilizes linear relationships between variables, and the proportionality of risks is always assumed. It is therefore worth exploring the capability of nonlinear models, which do not make any assumptions about the proportionality of risks.

The data used to test the model were selected from the database of the Liverpool Ocular Oncology Centre (Taktak et al., 2004). The dataset was split into two parts, one for training (1823 patterns), the other one for test (781 patterns). Nine prognostic factors were used: Sex, Tumour margin, Largest Ultrasound Basal Diameter, Extraocular extension, Presence of epithelioid cells, Presence of closed loops, Mitotic rate, Monosomy of chromosome 3.

The performance of survival analysis models can in general be assessed according to their discrimination and calibration aspects. Discrimination is the ability of the model to separate correctly the subjects into different groups. Calibration is the degree of correspondence between the estimated probability produced by the model and the actual observed probability (Dreiseitl & Ohno-Machado, 2002). One of the most widely used methods for assessing discrimination in survival analysis is Harrell's C index (Dreiseitl & Ohno-Machado, 2002), (Harrell et al. 1982), an extension to survival analysis of the Area Under the Receiver Operator Characteristic (AUROC). Calibration is assessed by a Kolmogorov-Smirnov (KS) goodness-of-fit test with corrections for censoring (Koziol, 1980).

The C index was evaluated for a set of years that are of interest to applications, from 1 to 7. The minimum was achieved at 7 years (0.75), the maximum at 1 year (0.8). The KS test with corrections for censoring was applied for the above set of years, and up to the maximum uncensored time (16.8 years); the confidence level was set as usual at 0.05. The null hypothesis that

the modeled distributions follow the empirical estimate cannot be rejected for years 1 to 7, whereas it is rejected if we compare the distributions up to 16.8 years; the null hypothesis is always rejected for the Cox model.

FUTURE TRENDS

Neural networks are very flexible modelling tools, and in the context of survival analysis they can offer advantages with respect to the (usually linear) modelling approaches commonly found in literature. This flexibility, however, comes at a cost: computational time and difficulty of interpretation of the model. The first aspect is due to the typically large number of parameters which characterise moderately complex networks, and the fact that the learning process results in a nonconvex, nonlinear optimization problem.

The second aspect is in some way a result of the nonlinearity and nonconvexity of the model. Addressing the issue of nonconvexity may be the first step to obtain models which can be easily interpreted in terms of their parameters, and easier to train; and in this respect, kernel machines (like Support Vector Machines) might be considered as the next step in flexible nonlinear modelling, although the formulation of learning algorithms for these models follows a paradigm which is not based on likelihood functions, and therefore their application to survival data is not immediate.

CONCLUSION

This article proposes a new neural network model for survival analysis in a continuous time setting, which approximates the logarithm of the hazard rate function. The model formulation allows an easy derivation of error bars on both hazard rate and survival predictions. The model is trained in the Bayesian framework to increase its robustness and to reduce the risk of overfitting. The model has been tested on real data, to predict survival from intraocular melanoma.

Formal discrimination and calibration tests have been performed, and the model shows good performance within a time horizon of 7 years, which is found useful for the application at hand.

C

Intelligence for Biopattern Analysis is Support of eHealthcare; URL:www.biopattern.org

REFERENCES

Bakker, B., Heskes, T. (1999). "A neural-Bayesian approach to survival analysis". *Proceedings IEE Artificial Neural Networks*, pp. 832-837.

Biganzoli, E., Boracchi, P., Mariani, L., Marubini, E. (1998). "Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach". *Statistics in Medicine*. 17, pp. 1169-86.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press Inc. New York.

Cox, D. R., Oakes, D. (1984). *Analysis of Survival Data*. Chapman and Hall.

Dreiseitl, S., Ohno-Machado, L. (2002). "Logistic regression and artificial neural network classification models: a methodology review". *Journal of Biomedical Informatics*, vol. 35, no. 5-6, pp. 352-359.

Eleuteri, A. , Tagliaferri, R. , Milano, L. , De Placido, S. , De Laurentiis, M. (2003). "A novel neural network-based survival analysis model", *Neural Networks*, 16, pp. 855-864.

Harrell Jr., F. E. , Califf, R. M. , Pryor, D. B. , Lee, K. L. and Rosati, R. A. (1982). "Evaluating the yield of medical tests". *Journal of the American Medical Association*, vol. 247, no. 18, pp. 2543-2546.

Koziol, J. (1980). "Goodness-of-fit tests for randomly censored data". *Biometrika* 67 (3), pp. 693-696.

Lisboa, P. J. G. , Wong, H. , Harris, P. , Swindell, R. (2003). "A Bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer". *Artificial intelligence in medicine*, 28, pp. 1-25.

MacKay, D. J. C. (1992). "The evidence framework applied to classification networks". *Neural Computation*, 4 (5), pp. 720-36.

Neal, R. M. (2001). *Survival Analysis Using a Bayesian Neural Network*. Joint Statistical Meetings report, Atlanta.

Ripley, B. D. , Ripley, R. M. (1998). "Neural Networks as Statistical Methods in Survival Analysis". *Artificial Neural Networks: Prospects for Medicine* (R. Dybowsky and V. Gant eds.), Landes Biosciences Publishers.

Schwarzer, G. , Vach, W. , Schumacher, M. (2000). "On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology". *Statistics in medicine* 19, pp. 541-561.

Taktak, A. F. G. , Fisher, A. C. , Damato, B. (2004). "Modelling survival after treatment of intraocular melanoma using artificial neural networks and Bayes theorem". *Physics in Medicine and Biology*. 49, pp. 87-98.

Williams, C. K. I. , Qazaz, C. , Bishop, C. M. , Zhu, H. (1995). "On the relationship between Bayesian error bars and the input data density". *Proceedings of the 4th International Conference on Artificial Neural Networks*, pp. 160-165, Cambridge (UK).

KEY TERMS

Bayesian Inference: Inference rules which are based on application of Bayes' theorem and the basic laws of probability calculus.

Censoring: Mechanism which precludes observation of an event. A form of missing data.

Hyperparameter: Parameter in a hierarchical problem formulation. In Bayesian inference, the parameters of a prior.

Neural Networks: A graphical representation of a nonlinear function. Usually represented as a directed acyclic graph. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Posterior Distribution: Probabilistic representation of knowledge, resulting from combination of prior knowledge and observation of data.

Prior Distribution: Probabilistic representation of prior knowledge.

Random Variable: Measurable function from a sample space to the measurable space of possible values of the variable.

C

Survival Analysis: Statistical analysis of data represented in terms of realisation of point events. In medical applications usually the point event is the death of an individual, or recurrence of a disease.

Configuration

Luca Anselma
Università di Torino, Italy

Diego Magro
Università di Torino, Italy

INTRODUCTION

Configuring means selecting and bringing together a set of given components to produce an aggregate (or a set of aggregates) satisfying some requirements. All the component types are predefined and no new component type may be created during the configuration process.

The result of the configuration can be physical objects (such as cars or elevators), non-physical entities (such as compound services or processes) or heterogeneous wholes made of both physical and non-physical parts (such as computer systems with their hardware and software components).

The configuration process has to take into consideration both endogenous and exogenous constraints: the former pertain to the type of the assembled object(s) (therefore they hold for all the individuals of that type) and mainly come from the interactions among components, whereas the latter usually represent requirements that the final aggregate(s) should satisfy. All these constraints can be very complex and make the manual solution of configuration problems a very hard task in many cases.

The complexity of configuration and its relevance in several application domains have stimulated the interest in its automation. Since the beginning, Artificial Intelligence has provided various effective techniques to achieve this goal. One of the first configurators was also one of the first commercially successful expert systems: a production rule-based system called *R1* (McDermott, 1982, 1993). *R1* was developed in the early Eighties to configure VAX computer systems, and it has been used for several years by Digital Equipment Corporation.

Since then, configuration has gained importance both in industry and in marketing, also due to both the support that it offers to the mass customization business strategy and the new commercial opportunities provided by the Web. Configuration is currently an important application field for many Artificial Intelligence techniques and it is still posing many interesting problems to scientific research.

BACKGROUND

The increasing complexity and size of configurable products made it clear that production-rule-based configurators such as R1 are not effective, particularly in the phase of maintenance of knowledge bases. In fact, changing a rule may require, as a side effect, changing several other rules and so on, and, actually, for some products, the component library may change frequently.

To partly address this problem, in current configurator systems, domain knowledge and control knowledge for problem solving are separate. The domain knowledge is represented in a declarative language, and the control knowledge (i.e., inferential mechanisms) is general (i.e., not depending on the particular problem to be solved). This is a common approach in modern knowledge-based systems. A configurator is based on an explicit representation of the general model of the configurable entities, which implicitly represents all the valid product individuals. The reasoning mechanisms implement the control knowledge and they use the domain knowledge to draw inferences and to compute configurations.

Regarding domain knowledge, there is a general agreement about what the concepts to represent are. In (Soininen, Tiihonen, Männistö & Sulonen, 1998) the authors introduce a widely accepted ***conceptualization*** for configuration problems. This conceptualization includes the concepts of

- *components*, which are the constituents of configurations;

- *parts* to describe the compositional structure;
- *ports* to model connections and compatibilities between components;
- *resources* that are produced, used or consumed by components;
- *functions* to represent functionalities;
- *attributes* used to describe components, ports, resources and functions;

- *taxonomies* in which component, port, resource and function types may be organized in;
- *constraints* to specify conditions that configurations must satisfy.

Figure 1 depicts a simplified fragment of the domain knowledge for PC configuration. It describes all the PC variants valid for the domain. Has-part relations

Figure 1. A fragment of a PC configuration knowledge base
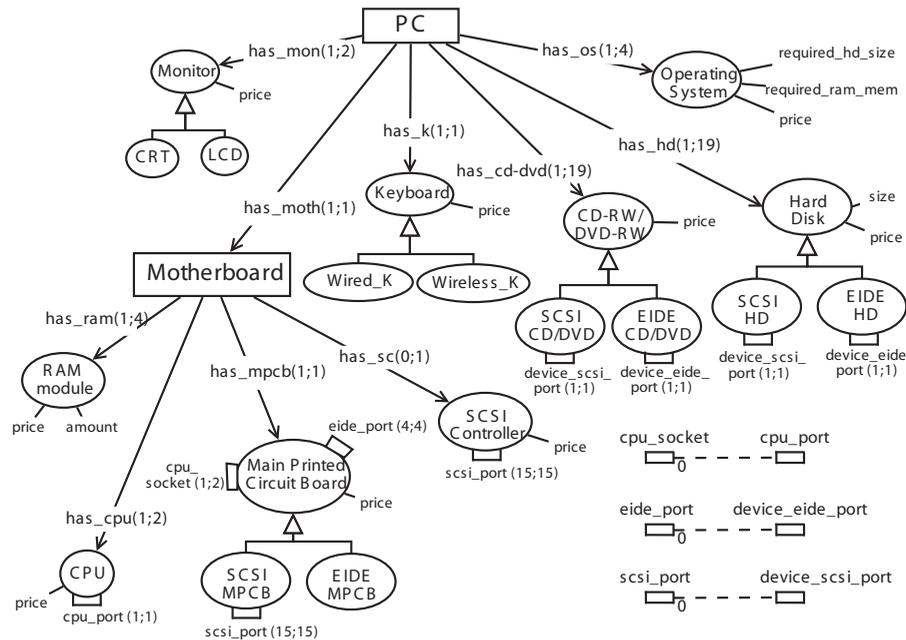
Constraints
- In any PC, if there is any SCSI device, then there must be either a SCSI Main Printed Circuit Board or a SCSI Controller
- In any PC, there must be no more than four EIDE devices and no more than fifteen SCSI devices
- In any PC, the total hard disk space required by all the Operating Systems must be less than the size of hard disks
- In any PC, the RAM required by each Operating System must be less than the available RAM amount
- In any Motherboard, there cannot be both a SCSI Main Printed Circuit Board and a SCSI Controller
- …

Notation
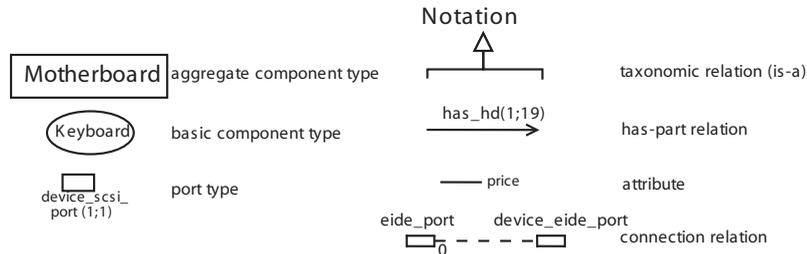
model the compositional structure of PCs (e.g., each PC has one or two monitors, a motherboard, etc.). Each component of a PC can be either of a basic (non configurable) type (e.g., the monitor) or of an aggregate (possibly configurable) type (e.g., the motherboard). Some relevant taxonomic relations are reported (e.g., the hard disks are either SCSI or EIDE). The basic components can be connected through ports (only few ports are reported): each port connects with at most one other port; for some ports the connection is optional (e.g., for eide_port), for others it is mandatory (e.g., for device_eide_port). Some attributes (e.g., the price) describe the components. A set of constraints model the interactions among the components: e.g., the third constraint specifies that hard disks must provide enough space, which is a resource consumed by operating systems.

Figure 3 describes a particular PC variant, meeting the requirements stated in Figure 2 (containing also an optimization criterion on price).

Figure 2. An example of user requirements for a PC

Requirements

The PC should have:
- two SCSI Hard Disks and at least 160 GB of Hard Disk
- at least 1 GB RAM
- a Wireless Keyboard
- the cheapest price
- …

Figure 3. A configured PC, compliant with the domain knowledge in Figure 1 and meeting the requirements in Figure 2

398

C
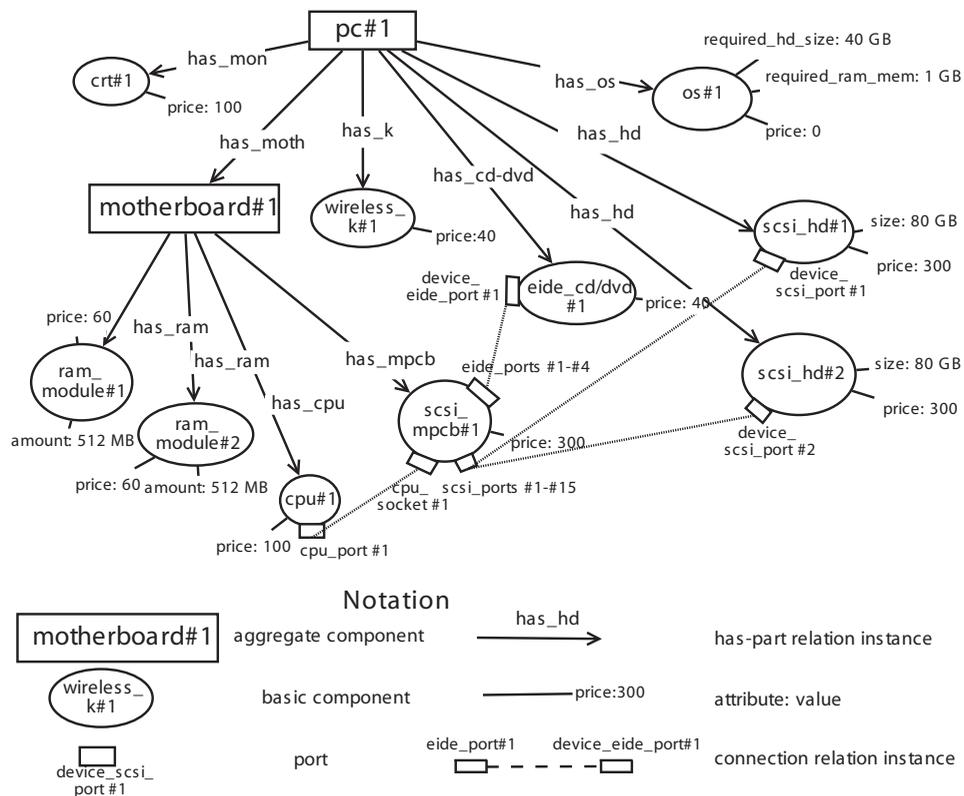
AUTOMATIC CONFIGURATION

Despite the consensus over the conceptualization of the problem, there is a wide range of approaches to configuration, with reference to different paradigms of Artificial Intelligence. It is possible to identify two mainstreams in current approaches to configuration: namely, constraint-based frameworks and logic-based frameworks. Constraint-based frameworks emphasize combinatorial aspects of configuration problems which have large search spaces and few solutions, while logic-based frameworks, in general, stress the description of the compositional structure of the product.

As regards *constraint-based frameworks*, approaches based on Constraint Satisfaction Problem (CSP) (Dechter, 2003) and its extensions are widely adopted. In particular, the classical CSP paradigm has been extended to overcome some of its limitations. In fact, on the one hand, in classical CSP, the set of variables is fixed and they will all be assigned values in every solution. On the other hand, in the configuration task the number and the types of the components that will be part of the final valid configuration are usually not known in advance, since they are selected by the configurator during the configuration process.

This fact motivated the introduction of Dynamic CSP (DCSP) (Mittal & Falkenhainer, 1990) (Gelle & Sabin, 2006) (also known as Conditional CSP) paradigm. A DCSP is defined – as classical CSP – on a fixed set of variables, but – differently from classical CSP – during the problem solving phase, it only takes into account the subset of variables relevant to the solution (i.e., the active variables). DCSP formalizes the notion of a particular type of constraint, i.e., activity constraints, which can add or remove variables from a potential solution depending on conditions imposed on already active variables. The search process starts with an initial set of active variables, and additional variables are introduced (or explicitly left out) as search progresses, depending on satisfied activity constraints.

A generalization of the original DCSP proposal, as well as some results on complexity and expressiveness, are presented in (Soininen, Gelle & Niemelä, 1999). Several solving methods for DCSP are described and discussed in (Gelle & Sabin, 2003).

In (Stumptner & Haselböck, 1993) the authors further extend DCSP by introducing the Generative CSP. In Generative CSP the types of the components may be compactly described and managed; moreover, generic constraints are defined: these are constraint schemata which can be instantiated on the specific variables activated at a particular point in the configuration process.

In (Sabin & Freuder, 1996) the authors overcome a second major limitation of CSP with regard to configuration problems: in fact, CSP is "flat", i.e., it does not allow to represent the structure of a configuration product in a straightforward way. To overcome this limitation, Sabin and Freuder propose Composite CSP, an extension to CSP which allows one to take into account not only changing sets of components, but also the hierarchical structure of the final configurations. In Composite CSP variables take not only atomic values but also values representing entire subproblems. Whenever a variable is assigned with a subproblem value, the subproblem is "expanded" and the problem is dynamically modified: specifically, it is "refined" by considering also the variables and the constraints in the subproblem. In such a way, it is easy to adapt the CSP's inferential mechanisms to Composite CSP.

Also classical CSP itself plays an important role in configuration. In fact, in (Aldanondo, Moynard & Hamou, 2000) an approach is presented that uses standard CSP techniques to solve configuration problems. Moreover, several results in the configuration research field somehow refer to standard CSP framework. For example, in (Freuder, Likitvivatanavong & Wallace, 2001) the authors explore the problem of generating explanations for configuration problems expressed as CSP. In (Freuder & O'Sullivan, 2001) the authors propose an approach for dealing with configuration problems expressed as CSP where it is not possible to satisfy all user requirements at the same time, and it is necessary to establish a satisfactory trade-off between them. (Amilhastre, Fargier & Marquis, 2002) extends CSP to offer support for interactive problem solving as in the case of interactive product configuration, where the interactivity refers to the user making choices during the configuration process. Specifically, the approach provides the user with features such as consistency maintenance (i.e., inconsistencies are discovered as soon as possible), consistency restoration (i.e., guidance for relaxing inconsistent choices) and explanations (i.e., minimal sets of inconsistent choices are identified). Finally, (Freuder, Carchrae & Beck, 2003) describes an approach for removing values of variables in a CSP that would lead to a dead-end in solving the CSP.

As regards *logic-based frameworks*, (McGuinness, 2002) analyzes Description Logics (DL) (Baader, Calvanese, McGuinness, Nardi & Patel-Schneider, 2003) as a convenient modeling tool for configurable products. DL make possible a description of the configuration knowledge by means of expressive conceptual languages with a rigorous semantics, thus enhancing knowledge comprehensibility and facilitating knowledge re-use. Furthermore, the powerful inference mechanisms currently available can be exploited both off-line by the knowledge engineers and on-line by the configuration system. Moreover, the paper describes a commercial DL-based family of configurators developed by AT&T.

(Soininen, Niemelä, Tiihonen & Sulonen, 2000) describes an approach in which the domain knowledge is represented with a high-level language and then mapped to a set of weight constraint rules, a form of logic programs offering support for expressing choices and both cardinality and resource constraints. Configurations are computed by finding stable Herbrand models of such a logic program.

(Sinz, Kaiser & Küchlin, 2003) presents an approach particularly geared to industrial context (in fact, it has been developed to be used by DaimlerChrysler for the configuration of their Mercedes lines). In Sinz et al.'s approach the domain knowledge is expressed as formulae in propositional logic; then, it is validated by running a satisfiability checker, which can also provide explanations in case of failure. However, this work aims at validating the knowledge base, rather than solving configuration problems.

There are also *hybrid approaches* that reconcile constraint-based frameworks and logic-based frameworks. For example, both (Magro & Torasso, 2003) and (Junker & Mailharro, 2003) describe hybrid frameworks based on a logic-based description of the structure of the configurable product (taking inspiration from logical languages derived from frame-based languages such as the DL) and on a constraint-based description of the possible ways of interaction between components.

In (Junker & Mailharro, 2003) constructs of DL are translated into concepts of constraint programming in order to solve a configuration problem. On the contrary, (Magro & Torasso, 2003) adopts an inference mechanism specific for configuration, which, basically, searches for tree-structured models on finite domains

for conceptual descriptions, and adapts some constraint-propagation techniques to the logical framework.

In most formalizations, the configuration task is theoretically intractable (at least NP-hard, in the worst case) and in some cases the intractability does appear also in practice and solving configuration problems can require a huge amount of CPU time. There are several ways that can be explored to cope with these situations: providing the configurator with a set of domain-specific heuristics, defining general focusing mechanisms (Magro & Torasso, 2001), making use of compilation techniques (Sinz, 2002) (Narodytska & Walsh, 2006), re-using past solutions (Geneste & Ruet, 2002), defining techniques to decompose a problem into a set of simpler subproblems (Magro, Torasso & Anselma, 2002) (Anselma & Magro, 2003).

Configuration has a growing commercial market. In fact, several configurator systems have been developed and some commercial tools are currently available (e.g., ILOG (Junker & Mailharro, 2003), Koalog, OfferIt! (Bergenti, 2004), Oracle, SAP (Haag, 2005), TACTON (Orsvarn, 2005)).

Furthermore, some Web sites have been equipped with configuration capabilities to support customers in selecting a suitable product in a wide range of domains such as cars (e.g., Porsche, Renault, Volvo), bikes (e.g., Pro-M Bike Configurator) and computers (e.g., Dell, Cisco).

FUTURE TRENDS

Many current configuration approaches and software configuration systems concern the configuration of mechanical or electronic devices/products and are conceived in order to be employed by domain experts, such as production or sales engineers.

Nowadays, the scope of configuration is growing and the application of automatic configuration techniques to non-physical entities is gaining more and more importance. The configuration of software products and complex services built on simpler ones are two research areas and application domains that are currently attracting the attention of researchers.

The capability of producing understandable *explanations* for their choices or for the inconsistencies that they encounter and of suggesting *consistency restorations* are some needs that configuration systems share with many knowledge-based or expert systems. However,

the aim of making configuration systems profitably used by non-expert users too and the deployment of configurators on the Web all contribute to strengthen the importance of these issues.

Explanations and restorations are also related to the topic of *interactive configuration*, which is still posing some challenging problems to researchers. Indeed, besides these capabilities, interactive configuration requires also effective mechanisms to deal with incomplete and/or incremental requirements specification (and with their retraction) and it is also demanding in terms of efficiency of the algorithms.

Real-world configuration knowledge bases can be very large and they usually are continually modified during their life cycle. Some research efforts are currently devoted to define powerful techniques and to design and implement tools that support *knowledge acquisition*, *knowledge-base verification* and *maintenance*.

Furthermore, a closer integration of configuration into the business models and of configurators into enterprise software systems is an important goal for several companies (as well as for enterprise software providers).

Distributed configuration is another important topic, especially in an environment where a specific complex product/service is provided by different suppliers that have to cooperate in order to produce it.

Finally, it is worth mentioning the *reconfiguration* of existing systems, which is still mainly an open problem.

CONCLUSION

Configuration has been a prominent area of Artificial Intelligence since the early Eighties, when it started to arouse interest among researchers working in academia and industry.

This article provides a general overview of the area of configuration by introducing the problem of configuration, briefly presenting a general conceptualization of configuration tasks, and succinctly describing some representative proposals in literature to deal with configuration problems.

As we have illustrated, during the last few years several approaches involving configuration techniques have been successfully applied in order to deal with issues pertaining to a wide range of real-world applica-

tion domains, ranging from cars to computer systems, from software to travel plans.

Theoretical results achieved by the academic environment have found effective, tangible applications in industrial settings, thus contributing to the diffusion of both industrial and commercial configurators. Such applications – in their turn – gave rise to new challenges, engendering a significant cross-fertilization of ideas among researchers in academia and in industry.

REFERENCES

Aldanondo, M., Moynard, G., & Hamou, K. H. (2000). General configurator requirements and modeling elements, *Proc. ECAI 2000 Configuration Workshop*, 1-6.

Amilhastre, J., Fargier, H., & Marquis, P. (2002). Consistency restoration and explanations in dynamic CSPs Application to configuration, *Artificial Intelligence* 135(1-2), 199-234.

Anselma, L., & Magro, D. (2003). Dynamic problem decomposition in configuration, *Proc. IJCAI 2003 Configuration Workshop*, 21-26.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (Editors) (2003). *The Description Logic Handbook*. Cambridge University Press.

Bergenti, F. (2004). Product and Servive Configuration for the Masses. *Proc. ECAI 2004 Configuration Workshop*, 7/1-7/6.

Dechter, R. (2003). *Constraint Processing*, Morgan Kaufmann.

Freuder, E., Carchrae, T., & Beck, J.C. (2003). Satisfaction Guaranteed. *Proc. IJCAI-03 Configuration Workshop*, 1-6.

Freuder, E., Likitvivatanavong, C., & Wallace, R. (2001). Explanation and implication for configuration problems, *Proc. IJCAI-01 Configuration Workshop*, 31-37.

Freuder, E., & O'Sullivan, B. (2001). Modeling and Generating Tradeoffs for Constraint-Based Configuration. *Proc. IJCAI-01 Configuration Workshop*, 38-44.

Gelle, E., & Sabin, M. (2003). Solving Methods for Conditional Constraint Satisfaction, *Proc. IJCAI-03 Configuration Workshop*, 7-12.

C

Gelle, E., & Sabin, M. (2006). Direct and Reformulation Solving of Conditional Constraint Satisfaction Problems . *Proc. ECAI-06 Configuration Workshop*, 14-19.

Geneste, L., & Ruet, M. (2002), Fuzzy Case Based Configuration, *Proc. ECAI 2002 Configuration Workshop*, 71-76.

Haag, A. (2005). "Dealing" with Configurable Products in the SAP Business Suite. *Proc. IJCAI-05 Configuration Workshop*, 68-71.

Junker, U., & Mailharro, D. (2003). The Logic of ILOG (J)Configurator: Combining Constraint Programming with a Description Logic. *Proc. IJCAI-03 Configuration Workshop*, 13-20.

Magro, D., & Torasso, P. (2001). Interactive Configuration Capability in a Sale Support System: Laziness and Focusing Mechanisms, *Proc. IJCAI-01 Configuration Workshop*, 57-63.

Magro, D., Torasso, P., & Anselma, L. (2002). Problem Decomposition in Configuration, *Proc. ECAI 2002 Configuration Workshop*, 50-55.

Magro, D., & Torasso, P. (2003). Decomposition Strategies for Configuration Problems, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *Special Issue on Configuration* 17(1), 51-73.

McDermott, J. (1982). R1: A Rule-Based Configurer of Computer Systems. *Artificial Intelligence* 19, 39-88.

McDermott, J. (1993). R1 ("XCON") at age 12: lessons from an elementary school achiever. *Artificial Intelligence* 59, 241-247.

McGuinness, D.L. (2002). Configuration. In Baader, F., McGuinness, D.L., Nardi, D. & Patel-Schneider, P.F. (Editors). *The Description Logic Handbook. Theory, implementation, and applications*. Cambridge University Press.

Mittal, S., & Falkenhainer, B. (1990). Dynamic Constraint Satisfaction Problems, *Proc. of the AAAI 90*, 25-32.

Narodytska, N., & Walsh, T. (2006). Constraint and Variable Ordering Heuristics for Compiling Configuration Problems, *Proc. ECAI 2006 Configuration Workshop*, 2-7.

Orsvarn, K. (2005). Tacton Configurator – Research directions, *Proc. IJCAI 2005 Configuration Workshop*, 75.

Sabin, D., & Freuder, E.C. (1996). Configuration as Composite Constraint Satisfaction, *Proc. Artificial Intelligence and Manufacturing. Research Planning Workshop*, 153-161.

Sinz, C. (2002). Knowledge Compilation for Product Configuration, *Proc. ECAI 2002 Configuration Workshop*, 23-26.

Sinz, C., Kaiser, A., & Küchlin, W. (2003). Formal methods for the validation of automotive product configuration data, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *Special Issue on Configuration* 17(1), 75-97.

Soininen, T., Gelle, E., & Niemelä, I. (1999). A Fixpoint Definition of Dynamic Constraint Satisfaction, *Lecture Notes in Computer Science* 1713, 419-433.

Soininen, T., Niemelä, I., Tiihonen, J., & Sulonen, R. (2000). Unified Configuration Knowledge Representation Using Weight Constraint Rules, *Proc. ECAI 2000 Configuration Workshop*, 79-84.

Soininen, T., Tiihonen, J., Männistö, T., & Sulonen, R. (1998). Towards a General Ontology of Configuration, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 12(4), 383-397.

Stumptner, M., & Haselböck, A. (1993). A Generative Constraint Formalism for Configuration Problems. *Lecture Notes in Artificial Intelligence* 728, 302-313.

KEY TERMS

Constraint Satisfaction Problem (CSP): A CSP is defined by a finite set of variables, where each variable is associated with a domain, and a set of constraints over a subset of variables, restricting the possible combinations of values that the variables in the subset may assume. A solution of a CSP is an assignment of a value to each variable that is consistent with the constraints.

Description Logics (DL): Logics that are designed to describe concepts and individuals in knowledge bases. They were initially developed to provide a precise semantics for the frame systems and the semantic net-

works. The typical inference for concepts is checking if a concept is more general than (i.e., subsumes) another one. The typical inference for individuals is checking if an individual is an instance of a concept. Many DL are fragments of first-order logic, while some of them go beyond first order.

Logic Program: A logic theory (possibly containing some extra-logic operators) that can be given a procedural meaning such that the process of checking if a formula is derivable in the theory can be viewed as a program execution.

Mass Customization: A business strategy that combines the mass production paradigm with product personalization. It is closely related to the modularity in product design. This design strategy makes it possible to adopt the mass production model for standard modules, facilitates the management of product families and variants and it leaves room for (various kinds and degrees of) personalization.

Production-Rule-Based System: A system where knowledge is represented by means of production rules. A production rule is a statement composed of conditions and actions. If data in working memory satisfy the conditions, the related actions can be executed, resulting in an update of the working memory.

Propositional Logic Formula Satisfiability: The task of checking whether it is possible to assign a truth value to every variable that occurs in a propositional formula, such that the truth value of the whole formula equals *true*.

Stable Herbrand Model: A minimal set of facts satisfying a logic program (theory). Each fact in the model is a variable-free atom whose arguments are terms exclusively built through function and constant symbols occurring in the program and whose predicate symbols occur in the program as well. Facts not appearing in the model are regarded as false.

Constraint Processing

Roman Barták
Charles University in Prague, Czech Republic

INTRODUCTION

Constraints appear in many areas of human endeavour starting from puzzles like crosswords (the words can only overlap at the same letter) and recently popular Sudoku (no number appears twice in a row) through everyday problems such as planning a meeting (the meeting room must accommodate all participants) till solving hard optimization problems for example in manufacturing scheduling (a job must finish before another job). Though all these problems look like being from completely different worlds, they all share a similar base – the task is to find values of decision variables, such as the start time of the job or the position of the number at a board, respecting given constraints. This problem is called a Constraint Satisfaction Problem (CSP).

Constraint processing emerged from AI research in 1970s (Montanary, 1974) when problems such as scene labelling were studied (Waltz, 1975). The goal of scene labelling was to recognize a type of line (and then a type of object) in the 2D picture of a 3D scene. The possible types were convex, concave, and occluding lines and the combination of types was restricted at junctions of lines to be physically feasible. This scene labelling problem is probably the first problem formalised as a CSP and some techniques developed for solving this problem, namely arc consistency, are still in the core of constraint processing. Systematic use of constraints in programming systems has started in 1980s when researchers identified a similarity between unification in logic programming and constraint satisfaction (Gallaire, 1985) (Jaffar & Lassez, 1987). Constraint Logic Programming was born. Today Constraint Programming is a separate subject independent of the underlying programming language, though constraint logic programming still plays a prominent role thanks to natural integration of constraints into a logic programming framework.

This article presents mainstream techniques for solving constraint satisfaction problems. These techniques stay behind the existing constraint solvers and their understanding is important to exploit fully the available technology.

BACKGROUND

Constraint Satisfaction Problem is formally defined as a triple: a finite set of decision variables, a domain of possible values, and a finite set of constraints restricting possible combinations of values to be assigned to variables. Although the domain can be infinite, for example real numbers, frequently, a finite domain is assumed. Without lost of generality, the finite domain can be mapped to a set of integers which is the usual case in constraint solvers. This article covers finite domains only. In many problems, each variable has its own domain which is a subset of the domain from the problem definition. Such domain can be formally defined by a unary constraint. We already mentioned that constraints restrict possible combinations of values that the decision variables can take. Typically, the constraint is defined over a subset of variables, its scope, and it is specified either extensionally, as a set of value tuples satisfying the constraint, or intentionally, using a logical or arithmetical formula. This formula, for example $A < B$, then describes which value tuples satisfy the constraint. A small example of a CSP is ($\{A, B, C\}, \{1, 2, 3\}, \{A < B, B < C\}$).

The task of constraint processing is to instantiate each decision variable by a value from the domain in such a way that all constraints are satisfied. This instantiation is called a *feasible assignment*. Clearly, the problem whether there exists a feasible assignment for a CSP is NP-complete – problems like 3SAT or knapsack problem (Garey & Johnson, 1979) can be directly encoded as CSPs. Sometimes, the core constraint satisfaction problem is accompanied by a so called objective function defined over (some) decision variables and we get a *Constrained Optimisation Problem*. Then the task is to select among the feasible assignments the assign-

ment that minimizes (or maximizes) the value of the objective function. This article focuses on techniques for finding a feasible assignment but these techniques can be naturally extended to optimization problems via a well-known branch-and-bound technique (Van Hentenryck, 1989).

There are several comprehensive sources of information about constraint satisfaction starting from journal surveys (Kumar, 1992) (Jaffar & Maher, 1996) through on-line tutorials (Barták, 1998) till several books. Van Hentenryck's book (1989) was a pioneering work showing constraint satisfaction in the context of logic programming. Later Tsang's book (1993) focuses on constraint satisfaction techniques independently of the programming framework and it provides full technical details of most algorithms described later in this article. Recent books cover both theoretical (Apt, 2003) and practical aspects (Marriott & Stuckey, 1998), provide good teaching material (Dechter, 2003) or in-depth surveys of individual topics (Rossi *et al.*, 2006). We should not forget about books showing how constraint satisfaction technology is applied in particular areas; scheduling problems play a prominent role here (Baptiste *et al.*, 2001) because constraint processing is exceptionally successful in this area.

CONSTRAINT SATISFACTION TECHNIQUES

Constraint satisfaction problems over finite domains are basically combinatorial problems so they can be solved by exploring the space of possible (partial or complete) instantiations of decision variables. Later in this section we will present the typical search algorithms used in constraint processing. However, it should be highlighted that constraint processing is not simple enumeration and we will also show how so called consistency techniques contribute to solving CSPs.

Systematic Search

Search is a core technology of artificial intelligence and many search algorithms have been developed to solve various problems. In case of constraint processing we are searching for a feasible assignment of values to variables where the feasibility is defined by the constraints. This can be done in a backtracking manner where we assign a value to a selected variable and check whether

the constraints whose scope is already instantiated are satisfied. In the positive case, we proceed to the next variable. In the negative case, we try another value for the current variable or if there are no more values we backtrack to the last instantiated variable and try alternative values there. The following code shows the skeleton of this procedure called historically *labelling* (Waltz, 1975). Notice that the consistency check may prune domains of individual variables, which will be discussed in the next section.

```
procedure labelling(V,D,C)
        if all variables from V are assigned then return V
        select not-yet assigned variable x from V
        for each value v from Dx do
                (TestOK,D') ← consistent(V,D,C∪{x=v})
                if TestOK=true then
                        R ← labelling(V,D',C)
                        if R ≠ fail then return R
        end for
        return fail
end labelling
```

The above backtracking mechanism is parameterized by variable and value selection heuristics that decide about the order of variables for instantiation and about the order in which the values are tried. While value ordering is usually problem dependent and problem-independent heuristics are not frequently used due to their computational complexity, there are popular problem-independent variable ordering heuristics. Variable ordering is based on a so called *first-fail principle* formulated by Haralick and Eliot (1980) which says that the variable whose instantiation will lead to a failure with the highest probability should be tried first. A typical instance of this principle is a *dom* heuristic which prefers variables with the smallest domain for instantiation. There exist other popular variable ordering heuristics (Rossi *et al.*, 2006) such as *dom+deg* or *dom/deg*, but their detail description is out scope of this short article.

Though the heuristics influence (positively) efficiency of search they cannot resolve all drawbacks of backtracking. Probably the main drawback is ignoring the information about the reason of constraint infeasibility. If the algorithm discovers that no value can be assigned to a variable, it blindly backtracks to the last instantiated variable though the reason of the conflict may be elsewhere. There exist techniques like backjumping that can detect the variable whose instantiation caused the problem and backtrack (backjump) to this

variable (Dechter, 2003). These techniques belong to a broader class of intelligent backtracking that shares the idea of intelligent recovery from the infeasibility. Though these techniques are interesting and far beyond simple enumeration, it seems better to prevent infeasibility rather than to recover from it (even in an intelligent way).

Domain Filtering and Maintaining Consistency

Assume variables A and B with domain {1, 2, 3} and a simple constraint A < B. Clearly, value 3 can never be assigned to A because there is no way to satisfy the constraint A < B if this value is used for A. Hence, this value can be safely removed from the domain of variable A and it does not need to be assumed during search. Similarly, value 1 can be removed from the domain of B. This process is called *domain filtering* and it is realised by a special procedure assigned to each constraint. Domain filtering is closely related to consistency of the constraint. We say that constraint C is *(arc) consistent* if for any value x in the domain of any variable in the scope of C there exist values in the domains of other variables in the scope of C such that the value tuple satisfies C. Such value tuple is called a *support* for x. Domain filtering attempts to make the constraint consistent by removing values which have no support.

Domain filtering can be applied to all constraints in the problem to remove unsupported values from the domains of variables and to make the whole problem consistent. Because the constraints are interconnected, it may be necessary to repeat the domain filtering of a constraint C if another constraint pruned the domain of variable in the scope of C. Basically the domain filtering is repeated until a fixed point is reached which removes the largest number of unsupported values. There exist several procedures to realise this idea (Mackworth, 1977), AC-3 schema is the most popular one:

```
procedure AC-3(V,D,C)
   Q ← C
   while non-empty Q do
      select c from Q
      D' ← c.FILTER(D)
      if any domain in D' is empty then return (fail,D')
      Q ← Q ∪ {c'∈C | ∃x∈var(c') D'x≠Dx} – {c}
      D ← D'
   end while
   return (true,D)
end AC-3
```

We did not cover the details of the filtering procedure here. In the simplest way, it may explore the consistent tuples in the constraint to find a support for each value. There exist more advanced techniques that keep some information between the repeated calls to the filter and hence achieve better time efficiency (Bessiere, 1994). Frequently, the filtering procedure exploits semantics of the constraint to realise filtering faster. For example, filtering for constraint A < B can be realised by removing from the domain of A all values greater than the maximal value of B (and similarly for B).

Let us return our attention back to search. Even if we make all the constraints consistent, it does not mean that we obtained a solution. For example, the problem ({A, B, C}, {1, 2, 3}, {A ≠ B, B ≠ C}) is consistent in the above-described sense, but it has no solution. Hence consistency techniques need to be combined with backtracking search to obtain a complete constraint solver. First, we make the constraints consistent. Then we start the backtracking search as described in the previous section and after each variable instantiation, we make the constraints consistent again. It may happen that during the consistency procedure some domain becomes empty. This indicates inconsistency and we can backtrack immediately. Because the consistency procedure removes inconsistencies from the not yet instantiated variables, it prevents future conflicts during search. Hence this principle is called *look ahead* opposite to look back techniques that focus on recovery from discovered conflicts. The whole process is also called *maintaining consistency during search* and it can be realised by substituting the consistent procedure in labelling by the procedure AC-3. Figure 1 shows a difference between simple backtracking (top) and the look-ahead technique (bottom) when solving a well known 4-queens problem. The task is to allocate a queen to each column of the chessboard in such a way that no two queens attack each other. Notice that the look-ahead solved the method after four attempts while the simple backtracking is still allocating the first two queens.

Clearly, the more inconsistencies one can remove, the smaller search tree needs to be explored. There exist stronger consistency techniques that assume several constraints together (rather that filtering each constraint separately, as we described above), but they are usually too computationally expensive and hence they are not used in each node of the search tree. Nevertheless, there also exists a compromise between stronger and efficient

Figure 1. Solving 4-queens problem using backtracking (top) and look-ahead (bottom) techniques; the crosses indicate positions forbidden by the current allocation of queens in the look-ahead method (values pruned by AC).
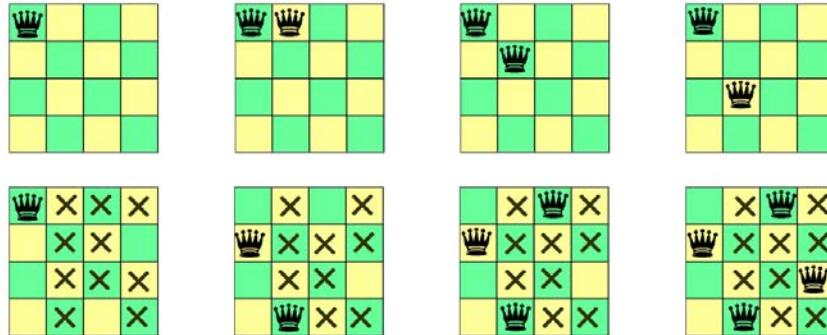
domain filtering called a *global constraint*. The idea is to encapsulate some well defined sub-problem into a single constraint (rather than a set of constraints) and then design a fast filtering algorithm for this constraint. A typical example of such global constraint is all-different that encapsulates a set of binary inequalities between all pairs of variables and by using filtering based on matching in bipartite graphs, it achieves stronger pruning (Régin, 1994). Figure 2 demonstrates how a CSP with binary inequalities is converted into a bipartite graph, where matching indicates a consistent instantiation of variables.

Global constraints represent a powerful mechanism how to integrate efficient solving algorithms into general framework of constraint satisfaction. There exist dozens of global constraints designed for particular application areas (Baptiste *et al*., 2001) as well as general global constraints (Beldiceanu *et al*., 2005).

FUTURE TRENDS

Constraint processing is a mature technology that goes beyond artificial intelligence and co-operates (and competes) with techniques from areas such as operations research and discrete mathematics. Many constraint satisfaction techniques including dozens of specialized as well as generic global constraints have been developed in recent years (Beldiceanu *et al*., 2005) and new techniques are coming. The technology trend is to integrate the techniques from different areas for co-operative and hybrid problem solving. Constraint processing may serve as a good base for such integration (as global constraints showed) but it can also provide solving techniques to be integrated in other frameworks such as SAT (satisfaction of logical formulas in a conjunctive normal form). This "hybridization and integration" trend is reflected in new conferences, for example

Figure 2. A graph representation of a constraint satisfaction problem with binary inequalities (left) and a bipartite graph representing the same problem in the all-different constraint (right).
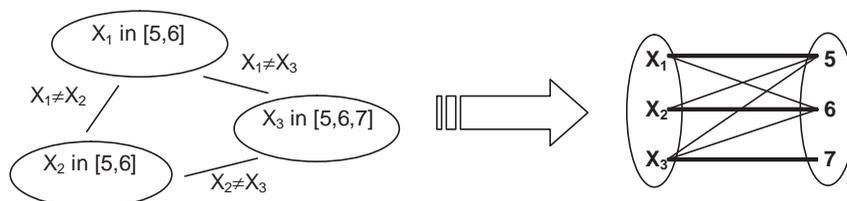
CP-AI-OR (International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems).

The paradox of fast technology development is that the technology is harder to use by non-expert users. There always exists several ways how to model a problem using constraints and though the models are equivalent concerning their soundness, they are frequently not equivalent concerning their efficiency. Although there are several rules of "good constraint modelling" (Marriott & Stuckey, 1998) (Barták, 2005) there do not exist generally applicable guidelines for constraint modelling. Hence it is sometimes not that easy to design a model that is solvable (in a reasonable time) by available constraint solvers. So one of the most important challenges of constraint processing for upcoming years is to bring the technology back to masses by providing automated modelling and problem reformulation tools that will form a middleware between the constraint solvers and non-expert users and make the holly grail of programming – the user states the problem and the computer solves it – a reality.

CONCLUSION

This article surveyed mainstream constraint satisfaction techniques with the goal to give a compact background of the technology to people who would like to use these techniques for solving combinatorial optimisation problems. We simplified the techniques and terminology a bit to fit the scope of the article while keeping the core principles. It is important to understand that the presented techniques (and even more) are already available in existing constraint solvers such us ILOG CP library (www.ilog.com/products/cp), SICStus Prolog (www.sics.se/sicstus), ECLiPSe (eclipse.crosscoreop. com), Mozart (www.mozart-oz.org), Choco (choco-solver.net) and other systems so the users are not required to program them from scratch. Nevertheless, understanding the underlying principles is important for design of efficient constraint models that can be solved by these systems. Constraint processing did not reach the holy grail of programming yet but it is going fast towards this goal.

REFERENCES

Apt, K. R. (2003). *Principles of Constraint Programming*. Cambridge University Press, Cambridge.

Baptiste, P.; Le Pape, C.; Nuijten, W. (2001). *Constraint-based Scheduling: Applying Constraints to Scheduling Problems.* Kluwer Academic Publishers, Dordrecht.

Barták, R. (1998). *On-line Guide to Constraint Programming*. Retrieved from the WWW: http://kti.mff. cuni.cz/~bartak/constraints.

Barták, R. (2005). Effective Modeling with Constraints. In *Applications of Declarative Programming and Knowledge Management*. LNAI 3392, Springer Verlag, Berlin pp. 149–165.

Beldiceanu, N.; Carlsson, M.; Rampon, J.X. (2005). Global constraint catalogue. Technical Report T2005-06, SICS, Uppsala.

Bessiere, C. (1994). Arc-consistency and arc-consistency again. *Artificial Intelligence* 65:179–190.

Dechter, R. (2003). *Constraint Processing.* Morgan Kaufmann, San Francisco.

Gallaire, H. (1985). Logic Programming: Further Developments. In *IEEE Symposium on Logic Programming*, IEEE, Boston, pp. 88–96.

Garey, M. R. & Johnson, D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco.

Haralick, R. M. & Elliot, G.L. (1980). Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence* 14:263–314.

Jaffar, J. & Lassez, J.L. (1987). Constraint Logic Programming. In *Proc. The ACM Symposium on Principles of Programming Language*s, ACM, Munich, pp. 111–119.

Jaffar, J. & Maher, M.J. (1996). Constraint Logic Programming – A Survey. *Journal of Logic Programming*, 19/20:503–581.

Kumar, V. (1992). Algorithms for Constraint Satisfaction Problems: A Survey. *AI Magazine* 13(1): 32–44.

Mackworth, A.K. (1977). Consistency in networks of relations. *Artificial Intelligence* 8:99–118.

Marriott, K. & Stuckey, P.J. (1998). *Programming with Constraints: An Introduction.* The MIT Press, Cambridge, MA.

Mohr, R. & Henderson, T.C. (1986). Arc and path consistency revised. *Artificial Intelligence* 28:225–233.

Montanari, U. (1974). Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences* 7:95–132.

Régin, J.-C. (1994). A filtering algorithm for constraints of difference in CSPs. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-94)*. AAAI Press, pp. 362–367.

Rossi, F.; Van Beek, P.; Walsh, T. (2006). *Handbook of Constraint Programming.* Elsevier, Oxford.

Tsang, E. (1993). *Foundations of Constraint Satisfaction.* Academic Press, London.

Van Hentenryck, P. (1989). *Constraint Satisfaction in Logic Programming*. The MIT Press, Cambridge, MA.

Waltz, D.L. (1975). Understanding line drawings of scenes with shadows. In *Psychology of Computer Vision*, McGraw-Hill, New York, pp. 19–91.

KEY TERMS

Constrained Optimisation Problem (COP): A Constraint Satisfaction Problem extended by an objective function over the (subset of) decision variables. The task is to find a solution to the CSP which minimizes or maximizes the value of the objective function.

Constraint: Any relation between a subset of decision variables. Can be expressed extensionally, as a set of value tuples satisfying the constraint, or intentionally, using an arithmetical or logical formula between the variables, for example $A+B < C$.

Constraint Satisfaction Problem (CSP): A problem formulated using a set of decision variables, their domains, and constraints between the variables. The task is to find an instantiation of decision variables by values from their domains in such a way that all the constraints are satisfied.

Consistency Techniques: Techniques that remove inconsistent values (from variables' domains) or value tuples, that is, the values that cannot be assigned to a given variable in any solution. Arc consistency is the most widely used consistency technique.

Decision Variable: A variable modelling some feature of the problem, for example a start time of activity, whose value we are looking for in such a way that specified constraints are satisfied.

Domain of Variable: A set of possible values that can be assigned to a decision variable, for example a set of times when some activity can start. Constraint processing usually assumes finite domains only.

Domain Pruning (Filtering): A process of removing values from domains of variables that cannot take part in any solution. Usually, due to efficiency issues only the values locally violating some constraint are pruned. It is the most common type of consistency technique.

Global Constraint: An n-ary constraint modelling a subset of simpler constraints by providing a dedicated filtering algorithm that achieves stronger or faster domain pruning in comparison to making the simpler constraints (locally) consistent. All-different is an example of a global constraint.

Look Ahead: The most common technique for integrating depth-first search with maintaining consistency. Each time a search decision is done, it is propagated in the problem model by making the model consistent.

Search Algorithms: Algorithms that explore the space of possible (partial or complete) instantiations of decision variables with the goal to find an instantiation satisfying all the constraints (and optimizing the objective function in case of COP).

C

Continuous ACO in a SVR Traffic Forecasting Model

Wei-Chiang Hong
Oriental Institute of Technology, Taiwan

INTRODUCTION

The effective capacity of inter-urban motorway networks is an essential component of traffic control and information systems, particularly during periods of daily peak flow. However, slightly inaccurate capacity predictions can lead to congestion that has huge social costs in terms of travel time, fuel costs and environment pollution. Therefore, accurate forecasting of the traffic flow during peak periods could possibly avoid or at least reduce congestion. Additionally, accurate traffic forecasting can prevent the traffic congestion as well as reduce travel time, fuel costs and pollution.

However, the information of inter-urban traffic presents a challenging situation; thus, the traffic flow forecasting involves a rather complex nonlinear data pattern and unforeseen physical factors associated with road traffic situations. Artificial neural networks (ANNs) are attracting attention to forecast traffic flow due to their general nonlinear mapping capabilities of forecasting. Unlike most conventional neural network models, which are based on the empirical risk minimization principle, support vector regression (SVR) applies the structural risk minimization principle to minimize an upper bound of the generalization error, rather than minimizing the training errors. SVR has been used to deal with nonlinear regression and time series problems. This investigation presents a short-term traffic forecasting model which combines SVR model with continuous ant colony optimization (SVRCACO), to forecast inter-urban traffic flow. A numerical example of traffic flow values from northern Taiwan is employed to elucidate the forecasting performance of the proposed model. The simulation results indicate that the proposed model yields more accurate forecasting results than the seasonal autoregressive integrated moving average (SARIMA) time-series model.

BACKGROUND

Traditionally, there has been a wide variety of forecasting approaches applied to forecast the traffic flow of inter-urban motorway networks. Those approaches could be classified according to the type of data, forecast horizon, and potential end-use (Dougherty, 1996); including historical profiling (Okutani & Stephanedes, 1984), state space models (Stathopoulos & Karlafits, 2003), Kalman filters (Whittaker, Garside & Lindveld, 1994), and system identification models (Vythoulkas, 1993). However, traffic flow data are in the form of spatial time series and are collected at specific locations at constant intervals of time. The above-mentioned studies and their empirical results have indicated that the problem of forecasting inter-urban motorway traffic flow is multi-dimensional, including relationships among measurements made at different times and geographical sites. In addition, these methods have difficultly coping with observation noise and missing values while modeling. Therefore, Danech-Pajouh and Aron (1991) employed a layered statistical approach with a mathematical clustering technique to group the traffic flow data and a separately tuned linear regression model for each cluster. Based on the multi-dimensional pattern recognition requests, such as intervals of time and geographical sites, non-parametric regression models (Smith, Williams & Oswald, 2002) have also successfully been employed to forecast motorway traffic flow. The ARIMA model and extended models are the most popular approaches in traffic flow forecasting (Kamarianakis & Prastacos, 2005) (Smith et al., 2002). Due to the stochastic nature and the strongly nonlinear characteristics of inter-urban traffic flow data, the artificial neural networks (ANNs) models have received much attention and been considered as alternatives for traffic flow forecasting models (Ledoux, 1997) (Yin, Wong, Xu & Wong, 2002). However, the training procedure of ANNs models is not only time consuming but also possible to get trapped in local minima and subjectively in selecting the model architecture.

C

Thus, SVR have been successfully employed to solve forecasting problems in many fields. Such as financial time series (stocks index and exchange rate) forecasting (Pai & Lin, 2005) (Pai, Lin, Hong & Chen, 2006), engineering and software field (production values and reliability) forecasting (**Hong &** Pai, 2006) (Pai & Hong, 2006), atmospheric science forecasting (Hong & Pai, 2007) (Mohandes, Halawani, Rehman & Hussain, 2004), and so on. Meanwhile, SVR model had also been successfully applied to forecast electric load (Pai & Hong, 2005a) (Pai & Hong, 2005b). The practical results indicated that poor forecasting accuracy is suffered from the lack of knowledge of the selection of the three parameters (σ, C, and ε) in a SVR model.

In this investigation, one of evolutionary algorithms, the ant colony optimization (ACO), is tried to determine the values of three parameters in a SVR traffic flow model in Panchiao city of Taipei County, Taiwan. In addition, as being developed for discrete optimization, the application of ACO to continuous optimization problems requires the transformation of a continuous search space to a discrete one by discretization of the continuous decision variables, which procedure is so-called CACO.

MAIN FOCUS OF THE CHAPTER

In this article, two models, the seasonal ARIMA (SARIMA) model and the SVRCACO model, are used to compare the forecasting performance of traffic flow.

Support Vector Regression (SVR) Model

The basic concept of the SVR is to map nonlinearly the original data x into a higher dimensional feature space. Hence, given a set of data $G = \{(x_i, a_i)\}_{i=1}^{N}$ (where x_i is the input vector; a_i is the actual value, and N is the total number of data patterns), the SVM regression function is:

$$f = g(x) = \mathbf{w}^{T}\phi(x_i) + b \qquad (1)$$

where $\phi(x_i)$ is the feature of inputs (to map the input data into a so-called high dimensional feature space, see **Fig. 1 (a)** and **(b)**), and both w and b are coefficients. The coefficients (w and b) are estimated by minimizing the following regularized risk function

$$R(f) = C\frac{1}{N}\sum_{i=1}^{N} L_e(a_i, f_i) + \frac{1}{2}\|w\|^2 \qquad (2)$$

where

$$L_\varepsilon(a, f) = \begin{cases} 0 & if \quad |a - f| \le \varepsilon \\ |a - f| - \varepsilon & otherwise \end{cases} \qquad (3)$$

In addition, $L_\varepsilon(a, f)$ is employed to find out an optimum hyper plane on the high dimensional feature space to maximize the distance separating the training data into two subsets. Thus, the SVR focuses on finding the optimum hyper plane and minimizing the training error between the training data and the ε-insensitive loss function (as thick line in **Fig. 1(c)**).

Minimize:

$$R(w, \xi, \xi^*) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{N}(\xi_i + \xi_i^*)\right) \qquad (4)$$

with the constraints,

$$w\phi(x_i) + b - a_i \le \varepsilon + \xi_i^*$$

$$a_i - w\phi(x_i) - b \le \varepsilon + \xi_i$$

$$\xi_i, \xi_i^* \ge 0$$

$$i = 1, 2, \dots, N$$

The first term of Eq. (5), employed the concept of maximizing the distance of two separated training data, is used to regularize weight sizes, to penalize large weights, and to maintain regression function flatness. The second term penalizes training errors of forecasting values and actual values by using the ε-insensitive loss function. C is a parameter to trade off these two terms. Training errors above ε are denoted as ξ_i^*, whereas training errors below ε are denoted as ξ_i.

After the quadratic optimization problem with inequality constraints is solved, the weight w in Eq. (2) is obtained,

$$w^* = \sum_{i=1}^{N}(\beta_i - \beta_i^*)K(x, x_i) \qquad (5)$$

Figure 1. Transformation process illustration of a SVR model
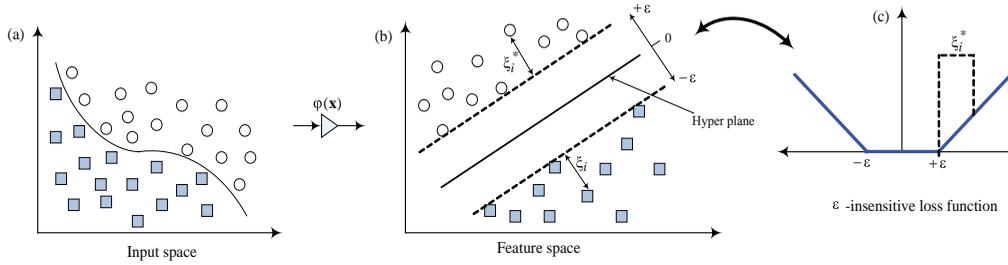
Hence, the regression function is Eq. (6):

$$g(x, \beta, \beta^*) = \sum_{i=1}^{N} (\beta_i - \beta_i^*) K(x, x_i) + b \qquad (6)$$

Here, $K(x_i, x_j)$ is called the Kernel function. The value of the Kernel equals the inner product of two vectors, x_i and x_j, in the feature space $\phi(x_i)$ and $\phi(x_j)$; that is, $K(x_i, x_j) = \phi(x_i) * \phi(x_j)$. The Gaussian RBF kernel is not only easier to implement, but also capable to nonlinearly map the training data into an infinite dimensional space, thus, it is suitable to deal with nonlinear relationship problems. In this work, the Gaussian function, $\exp(-\|x - x_i\|^2 / 2\sigma^2)$, is used in the SVR model.

CACO in Selecting Parameters of the SVR Model

Ant colony optimization algorithms (Dorigo, 1992) have been successfully used to dealing with combinatorial optimization problems such as job-shop scheduling (Colorni, Dorigo, Maniezzo & Trubian, 1994), traveling salesman problem (Dorigo & Gambardella, 1997), space-planning (Bland, 1999), quadratic assignment problems (Maniezzo & Colorni, 1999), and data mining (Parpinelli, Lopes & Freitas, 2002). ACO imitates the behaviors of real ant colonies as they forage for food, wherein each ant lays down the pheromone on the path to the food sources or back to the nest. The paths with more pheromone are more likely to be selected by other ants. Over time, a colony of ants will select the shortest path to the food source and back to the nest. Therefore, a pheromone trail is the most important process for individual ant to smell and select its route.

The probability, $P_k(i,j)$, that an ant k moves from city i to city j is expressed as Eq. (7),

$$P_k(i,j) = \begin{cases} \arg \max_{S \in M_k} \left\{ [\tau(i,S)]^\alpha [\eta(i,S)]^\beta \right\} & , \quad \text{if } q \leq q_0 \\ \text{Eq.(9)} & , \quad otherwise \end{cases} \qquad (7)$$

$$P_k(i,j) = \begin{cases} [\tau(i,j)]^\alpha [\eta(i,j)]^\beta \Big/ \sum_{S \in M_k} [\tau(i,S)]^\alpha [\eta(i,S)]^\beta & , \quad j \notin M_k \\ 0 & , \quad otherwise \end{cases} \qquad (8)$$

where $\tau(i,j)$ is the pheromone level between city i and city j, $\eta(i,j)$ is the inverse of the distance between cities i and j. In this study, the forecasting error represents the distance between cities. The α and β are parameters determining the relative importance of pheromone level and M_k is a set of cities in the next column of the city matrix for ant k. q is a random uniform variable [0,1] and the value q_0 is a parameter. The values of α, β and q_0 are set to be 8, 5 and 0.2 respectively.

Once ants have completed their tours, the most pheromone deposited by ants on the visited paths is considered as the information regarding the best paths from the nest to the food sources. Therefore, the pheromone dynamic updating plays the main role in real ant colonies searching behaviors. The local and global updating rules of pheromone are expressed as Eq.(9) and Eq(10) respectively.

$$\tau(i,j) = (1-\rho)\tau(i,j) + \rho\,\tau_0 \qquad (9)$$

$$\tau(i,j) = (1-\delta)\tau(i,j) + \delta\Delta\tau(i,j) \qquad (10)$$

C

where ρ is the local evaporation rate of pheromone, $0 < \rho < 1$; τ_0 is the initial amount of pheromone deposited on each of the paths. In this work, the values of ρ and τ_0 are set to be 0.01 and 1 correspondingly. In addition, the approach proposed by Dorigo and Gambardella (1994) was employed here for generating the initial amount of pheromone. Global trail updating is accomplished according Eq.(10). The δ is the global pheromone decay parameter, $0 < \delta < 1$, and set to be 0.2 for this study. The $\Delta\tau(i, j)$, expressed as Eq.(11), is used to increase the pheromone on the path of the solution.

$$\Delta\tau(i, j) = \begin{cases} 1/L & , \quad if\ (i, j) \in global\ best\ route \\ 0 & , \qquad\qquad otherwise \end{cases}$$

(11)

where L is the length of the shortest route.

A Numerical Example and Experimental Results

The traffic flow data sets were originated from three Civil Motorway detector sites. The Civil Motorway is the busiest inter-urban motorway networks in Panchiao city, the capital of Taipei County, Taiwan. The major site was located at the center of Panchiao City, where the flow intersects an urban local street system, and it provided one way traffic volume for each hour in weekdays. Therefore, one way flow data for peak traffic are employed in this investigation, which includes the morning peak period (MPP; from 6:00 to 10:00) and the evening peak period (EPP; from 16:00 to 20:00). The data collection is conducted from February 2005 to March 2005, the number of traffic flow data available for MPP and EPP are 45 and 90 hours, respectively. For convenience, the traffic flow data are converted to equivalent of passengers (EOP), and both of these two peak periods show the seasonality of traffic data. In addition, traffic flow data are divided into three parts: training data (MPP 25 hours; EPP 60 hours), validation data (MPP 10 hours; EPP 15 hours) and testing data (MPP 10 hours; EPP 15 hours). The accuracy of forecasting models is measured by the normalized root mean square error (NRMSE), as given by Eq.(12).

$$NRMSE = \sqrt{\sum_{i=1}^{n}(a_i - f_i)^2 \Big/ \sum_{i=1}^{n} a_i^2}$$

(12)

where n is the number of forecasting periods; a_i is the actual traffic flow value at period i; and f_i is the forecasting traffic flow value at period i.

The parameter selection of forecasting models is important for obtaining good forecasting performance. For the SARIMA model, the parameters are determined by taking the first-order regular difference and first seasonal difference to remove non-stationary and sea-sonality characteristics. Using statistical packages, with no residuals autocorrelated and approximately white noise residuals, the most suitable models for these two morning/evening peak periods for the traffic data are SARIMA$(1,0,1) \times (0,1,1)_5$ with non-constant item and SARIMA$(1,0,1) \times (1,1,1)_5$ with constant item, respectively. The equations used for the SARIMA models are presented as Eqs. (13) and (14), respectively.

$$(1 - 0.5167B)(1 - B^5)X_t = (1 + 0.3306B)(1 - 0.9359B^5)\varepsilon_t$$

(13)

$$(1 - 0.5918B)(1 - B^5)X_t = 2.305 + (1 - 0.9003B^5)\varepsilon_t$$

(14)

For the SVRCACO model, a rolling-based forecasting procedure was conducted and a one-hour-ahead forecasting policy adopted. Then, several types of data-rolling are considered to forecast traffic flow in the next hour. In this investigation, the CACO is employed to determine suitable combination of the three parameters in a SVR model. Parameters of the SVRCACO models with the minimum testing NRMSE values were selected as the most suitable model for this investigation. **Table 1** indicates that SVRCACO models perform the best when 15 and 35 input data are used for morning/evening traffic forecast respectively. **Table 2** compares the forecasting accuracy of the SARIMA and SVRCACO models in terms of NRMSE. It is illustrated that SVRCACO models have better forecasting results than the SARIMA models.

FUTURE TRENDS

In this investigation, the SVRCACO model provides a convenient and valid alternative for traffic flow forecasting. The SVRCACO model directly uses historical observations from traffic control systems and then determines suitable parameters by efficient optimiza-

Table 1. Forecasting results and associated parameters of the SVRCACO models

Morning peak period					Evening peak period				
Nos. of input data	Parameters			NRMSE of testing	Nos. of input data	Parameters			NRMSE of testing
	σ	*C*	ε			σ	*C*	ε	
5	0.7286	2149.2	0.8249	0.3965	25	0.7277	9658.9	0.4176	0.1112
10	0.7138	1199.0	0.1460	0.3464	30	0.9568	9337.7	0.7741	0.1037
15	**0.7561**	**2036.5**	**0.9813**	**0.2632**	**35**	**0.8739**	**6190.2**	**0.7619**	**0.1033**
20	0.6858	2141.6	0.4724	0.2754	40	0.1528	6300.5	0.8293	0.1147
					45	0.5093	3069.9	0.7697	0.1077
					50	0.1447	8835.7	0.8616	0.1247
					55	0.5798	6299.1	0.5796	0.1041

Table 2. Forecasting results (unit: EOP)

Morning peak period				Evening peak period			
Peak periods	Actual	SARIMA	SVRCACO	Peak periods	Actual	SARIMA	SVRCACO
031106	1,317.5	1,363.77	2,190.9	031016	2,310.5	2,573.84	2,229.1
031107	2,522.0	2,440.11	2,027.4	031017	2,618.0	2,821.57	2,319.9
031108	2,342.0	2,593.91	2,140.4	031018	2,562.0	3,107.01	2,300.8
031109	2,072.0	2,422.09	2,313.7	031019	2,451.5	3,103.66	2,571.6
031110	1,841.5	2,459.87	2,053.4	031020	2,216.5	3,011.80	2,447.2
031206	995.5	1,578.34	1,980.6	031116	2,175.5	2,611.58	2,432.4
031207	1,457.0	2,569.92	1,704.1	031117	2,577.0	2,859.31	2,169.4
031208	1,899.0	2,690.35	1,548.3	031118	2,879.5	3,144.75	2,450.4
031209	1,870.5	2,505.38	1,521.0	031119	2,693.0	3,141.40	2,598.4
031210	2,151.5	2,537.98	1,881.1	031120	2,640.0	3,049.54	2,671.9
				031216	2,146.5	2,649.32	2,628.5
				031217	2,544.5	2,897.05	2,633.1
				031218	2,873.0	3,182.49	2,538.0
				031219	2,567.5	3,179.13	2,670.8
				031220	2,660.5	3,087.28	2,562.7
NRMSE		0.3039	0.2632	NRMSE		0.1821	0.1033

*: "031106" denotes the 6 o'clock on 11 March 2005, and so on.

tion algorithms. In future research, other factors and meteorological control variables during peak periods, such as driving speed limitation, important social events, the percentage of heavy vehicles, bottleneck service level and waiting time during intersection traffic signals can be included in the traffic forecasting model. In addition, some other advanced optimization algorithms for parameters selection can be applied for the SVR model to satisfy the requirement of real-time traffic control systems.

CONCLUSION

Accurate traffic forecast is crucial for the inter-urban traffic control system, particularly for avoiding congestion and for increasing efficiency of limited traffic resources during peak periods. The historical traffic data of Panchiao City in northern Taiwan shows a seasonal fluctuation trend which occurs in many inter-urban traffic systems. Therefore, over-prediction or under-prediction of traffic flow influences the transportation capability of an inter-urban system. This study introduces the application of forecasting techniques, SVRCACO, to investigate its feasibility for forecasting inter-urban motorway traffic. This article indicates that the SVR-CACO model has better forecasting performance than the SARIMA model. The superior performance of the SVRCACO model is due to the generalization ability of SVR model for forecasting and the proper selection of SVR parameters by CACO.

REFERENCES

1. Bland, J. A. (1999). Space-Planning by Ant Colony Optimization. *International Journal of Computer Applications in Technology*. (12) 320-328.

2. Box, G. E. P., & Jenkins, G. M. (1970). *Time Series Analysis, Forecasting and Control*. San Francisco: Holden-Day.

3. Colorni, A., Dorigo, M., Maniezzo, V., & Trubian, M. (1994). Ant System For Job-Shop Scheduling. *JORBEL—Belgian Journal of Operations Research Statistics and computer Science*. (34) 39-53.

4. Danech-Pajouh, M., & Aron, M. (1991). ATHENA: A Method For Short-Term Inter-Urban Motorway Traffic Forecasting. *Recherche Transports Sécurité (English Issue)*. (6) 11-16.

5. Dorigo, M. (1992). *Optimization, Learning, and Natural Algorithms*. Ph.D. Thesis, Dip. Elettronica e Informazione, Politecnico di Milano, Italy.

6. Dorigo, M., & Gambardella, L. (1997). Ant Colony System: A cooperative Learning Approach to The Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*. (1) 53-66.

7. Dougherty, M. S. (1996). *Investigation of Network Performance Prediction Literature Review*. Technical Note, Vol. 394, Institute for Transport Studies, University of Leeds.

8. Hong, W. C., & Pai, P. F. (2006). Predicting engine reliability by support vector machines. *International Journal of Advanced Manufacturing Technology*. (28) 154-161.

9. Hong, W. C., & Pai, P. F. (2007). Potential assessment of the support vector regression technique in rainfall forecasting. *Water Resources Management*. (21) 495-513.

10. Kamarianakis, Y., & Prastacos, P. (2005). Space-Time Modeling of Traffic Flow. *Computers & Geosciences*. (31) 119-133.

11. Ledoux, C. (1997). An Urban Traffic Flow Model Integrating Neural Networks. *Transportation Research Part C*. (5) 287-300.

12. Maniezzo, V., & Colorni, A. (1999). The Ant System Applied to The Quadratic Assignment Problem. *IEEE Transactions on Knowledge and Data Engineering*. (11) 769-778.

13. Mohandes, M. A., Halawani, T. O., Rehman, S., & Hussain, A. A. (2004). Support vector machines for wind speed prediction. *Renewable Energy*. (29) 939-947.

14. Okutani, I., & Stephanedes, Y. J. (1984). Dynamic Prediction of Traffic Volume Through Kalman Filtering Theory. *Transportation Research Part B*. (18) 1-11.

15. Pai, P. F., & Lin, C. S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*. (33) 497-505.

16. Pai, P. F., Lin, C. S., Hong, W. C., & Chen, C. T. (2006). A hybrid support vector machine regression

for exchange rate prediction. *International Journal of Information and Management Sciences*. (17) 19-32.

17. Pai, P. F., & Hong, W. C. (2006). Software reliability forecasting by support vector machines with simulated annealing algorithms. *Journal of Systems and Software*. (79) 747-755.

18. Pai, P. F., & Hong, W. C. (2005a). Forecasting regional electric load based on recurrent support vector machines with genetic algorithms. *Electric Power Systems Research*. (74) 417-425.

19. Pai, P. F., & Hong, W. C. (2005b). Support vector machines with simulated annealing algorithms in electricity load forecasting. *Energy Conversion and Management*. (46) 2669-2688.

20. Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). Data Mining With An Ant Colony Optimization Algorithm. *IEEE Transactions on Evolutionary Computation*. (6) 321-332.

21. Smith, B. L., Williams, B. M., & Oswald, R. K. (2002). Comparison of Parametric and Nonparametric Models For Traffic Flow Forecasting. *Transportation Research Part C*. (10) 303-321.

22. Stathopoulos, A., & Karlaftis, G. M. (2003). A Multivariate State Space Approach for Urban Traffic Flow Modeling and Prediction. *Transportation Research Part C*. (11) 121-135.

23. Vythoulkas, P. C. (1993). Alternative Approaches to Short Term Forecasting For Use in Driver Information Systems. *Transportation and Traffic Theory*, Amsterdam: Elsevier.

24. Whittaker, J., Garside, S., & Lindveld, K. (1994). Tracking and Predicting A Network Traffic Process. *Proceedings of the Second DRIVE II Workshop on Short-Term Traffic Forecasting*, Delft.

25. Yin, H., Wong, S. C., Xu, J., & Wong C. K. (2002). Urban Traffic Flow Prediction Using A Fuzzy-Neural Approach. *Transportation Research Part C*. (10) 85-98.

KEY TERMS

Ant Colony Optimization Algorithm (ACO): inspired by the behavior of ants in finding paths from the colony to food, is a probabilistic technique for solving computational problems which can be reduced to
416

finding good paths through graphs. A short path gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate.

Artificial Neural Networks (ANNs): A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data.

Autoregressive Integrated Moving Average (ARIMA): A generalization of an autoregressive moving average (ARMA) model. These models are fitted to time series data either to better understand the data or to predict future points in the series. The model is generally referred to as an ARIMA(p,d,q) model where p, d, and q are integers greater than or equal to zero and refer to the order of the autoregressive, integrated, and moving average parts of the model respectively.

Evolutionary Algorithm (EA): is a generic population-based meta-heuristic optimization algorithm. An EA uses some mechanisms inspired by biological evolution: reproduction, mutation, recombination, natural selection and survival of the fittest. Evolutionary algorithms consistently perform well approximating solutions to all types of problems because they do not make any assumption about the underlying fitness landscape.

Pheromone: A pheromone is a chemical that triggers an innate behavioral response in another member of the same species. There are alarm pheromones, food trail pheromones, sex pheromones, and many others that affect behavior or physiology. In this article, food trail pheromones are employed, which are common in social insects.

Seasonal Autoregressive Integrated Moving Average (SARIMA): A kind of ARIMA model to conduct forecasting problem while seasonal effect is suspected. For example, consider a model of daily road traffic volumes. Weekends clearly exhibit different behavior from weekdays. In this case it is often considered better to use a SARIMA (seasonal ARIMA) model than to increase the order of the AR or MA parts of the model.

Support Vector Machines (SVMs): Support vector machines (SVMs) were originally developed to solve pattern recognition and classification problems. With

the introduction of Vapnik's ε-insensitive loss function, SVMs have been extended to solve nonlinear regression estimation problems which are so-called support vector regression (SVR). SVR applies the structural risk minimization principle to minimize an upper bound of the generalization error. SVR has been used to deal with nonlinear regression and time series problems.

Data Mining Fundamental Concepts and Critical Issues

John Wang
Montclair State University, USA

Qiyang Chen
Montclair State University, USA

James Yao
Montclair State University, USA

INTRODUCTION

Data mining is the process of extracting previously unknown information from large databases or data warehouses and using it to make crucial business decisions. Data mining tools find patterns in the data and infer rules from them. The extracted information can be used to form a prediction or classification model, identify relations between database records, or provide a summary of the databases being mined. Those patterns and rules can be used to guide decision making and forecast the effect of those decisions, and data mining can speed analysis by focusing attention on the most important variables.

BACKGROUND

We are drowning in data, but starving for knowledge. In recent years the amount or the volume of information has increased significantly. Some researchers suggest that the volume of information stored doubles every year. Disk storage per person (DSP) is a way to measure the growth in personal data. Edelstein (2003) estimated that the number has dramatically grown from 28MB in 1996 to 472MB in 2000.

Data mining seems to be the most promising solution for the dilemma of dealing with too much data having very little knowledge. By using pattern recognition technologies and statistical and mathematical techniques to sift through warehoused information, data mining helps analysts recognize significant facts, relationships, trend, patterns, exceptions and anomalies. The use of data mining can advance a company's position by creating a sustainable competitive advantage. Data

warehousing and mining is the science of managing and analyzing large datasets and discovering novel patterns (Davenport & Harris, 2007; Wang, 2006; Olafsson, 2006).

Data mining is taking off for several reasons: organizations are gathering more data about their businesses, the enormous drop in storage costs, competitive business pressures, a desire to leverage existing information technology investments, and the dramatic drop in the cost/performance ratio of computer systems. Another reason is the rise of data warehousing. In the past, it was often necessary to gather the data, cleanse it, and merge it. Now, in many cases, the data are already sitting in a data warehouse ready to be used.

Over the last 40 years, the tools and techniques to process data and information have continued to evolve from data bases to data warehousing and further to data mining. Data warehousing applications have become business-critical. Data mining can compress even more value out of these huge repositories of information. Data mining is a multidisciplinary field covering a lot of disciplines such as databases, statistics, artificial intelligence, pattern recognition, machine learning, information theory, control theory, operations research, information retrieval, data visualization, high-performance computing or parallel and distributed computing, etc (Zhou, 2003; (Hand, Mannila, & Smyth, 2001).

Certainly, many statistical models had emerged a long time ago. Machine learning has marked a milestone in the evolution of computer science. Although data mining is still in its infancy, it is now being used in a wide range of industries and for a range of tasks in a variety of contexts (Wang, 2003; Lavoie, Dempsey, & Connaway, 2006). Data mining is synonymous with knowledge discovery in databases, knowledge extrac-

tion, data/pattern analysis, data archeology, data dredging, data snooping, data fishing, information harvesting, and business intelligence (Han and Kamber, 2001).

MAIN FOCUS

Functionalities and Tasks

The common types of information that can be derived from data mining operations are associations, sequences, classifications, clusters, and forecasting. Associations happen when occurrences are linked in a single event. One of the most popular association applications deals with market basket analysis. This technique incorporates the use of frequency and probability functions to estimate the percentage chance of occurrences. Business strategists can leverage off of market basket analysis by applying such techniques as cross-selling and up-selling. In sequences, events are linked over time. This is particularly applicable in e-business for Website analysis.

Classification is probably the most common data mining activity today. It recognizes patterns that describe the group to which an item belongs. It does this by examining existing items that already have been classified and inferring a set of rules from them. Clustering is related to classification, but differs in that no groups have yet been defined. Using clustering, the data-mining tool discovers different groupings within the data. The resulting groups or clusters help the end user make some sense out of vast amounts of data (Kudyba, & Hoptroff, 2001). All of these applications may involve predictions. The fifth application type, forecasting, is a different form of prediction. It estimates the future value of continuous variables based on patterns within the data.

Algorithms and Methodologies

Neural Networks

Also referred to as artificial intelligence (AI), neural networks utilize predictive algorithms. This technology has many similar characteristics to that of regression because the application generally examines historical data, and utilizes a functional form that best equates explanatory variables and the target variable in a man-

ner that minimizes the error between what the model had produced and what actually occurred in the past, and then applies this function to future data. Neural networks are a bit more complex as they incorporate intensive program architectures in attempting to identify linear, non-linear and patterned relationships in historical data.

Decision Trees

Megaputer (2006) mentioned that this method can be applied for solution of classification tasks only. As a result of applying this method to a training set, a hierarchical structure of classifying rules of the type "if…then…" is created. This structure has a form of a tree. In order to decide to which class an object or a situation should be assigned one has to answer questions located at the tree nodes, starting from the root. Following this procedure one eventually comes to one of the final nodes (called leaves), where the analyst finds a conclusion to which class the considered object should be assigned.

Genetic Algorithms (or Evolutionary Programming)

Genetic algorithms, biologically inspired search method, borrow mechanisms of inheritance to find solutions. Biological systems demonstrated flexibility, robustness and efficiency. Many biological systems are good at adapting to their environments. Some biological methods (such as reproduction, crossover and mutation) can be used as an approach to computer-based problem solving. An initial population of solutions is created randomly. Only a fixed number of candidate solutions are kept from one generation to the next. Those solutions that are less fit tend to die off, similar to the biological notion of "survival of the fittest".

Regression Analysis

This technique involves specifying a functional form that best describes the relationship between explanatory, driving or independent variables and the target or dependent variable the decision maker is looking to explain. Business analysts typically utilize regression to identify the quantitative relationships that exist between variables and enable them to forecast into the future.

Regression models also enable analysts to perform "what if" or sensitivity analysis. Some examples include how response rates change if a particular marketing or promotional campaign is launched, or how certain compensation policies affect employee performance and many more.

Logistics Regression

Logistic regression should be used when you want to predict the outcome of a dichotomous (e.g., yes/no) variable. This method is used for data that is not normally distributed (bell-shaped curve) i.e., categorical (coded) data. When a dependent variable can only have one of two answers, such as "will graduate" or "will not graduate", you cannot get a normal distribution as previously discussed.

Memory Based Reasoning (MBR) or the Nearest Neighbor Method

To forecast a future situation, or to make a correct decision, such systems find the closest past analogs of the present situation and choose the same solution which was the right one in those past situations. The drawback of this application is that there is no guarantee that resulting clusters provide any value to the end user. Resulting clusters may just not make any sense with regards to the overall business environment. Because of limitations of this technique, no predictive, "what if" or variable/target connection can be implemented.

The key differentiator between classification and segmentation with that of regression and neural network technology mentioned above is the inability of the former to perform sensitivity analysis or forecasting.

Applications and Benefits

Data mining can be used widely in science and business areas for analyzing databases, gathering data and solving problems. In line with Berry and Linoff (2004), the benefits data mining can provide for businesses are limitless. Here are just a few examples:

- *Identify best prospects and then retain them as customers.*
  By concentrating marketing efforts only on the best prospects, companies will save time and money,

thus increasing effectiveness of their marketing operation.

- *Predict cross-sell opportunities and make recommendations.*
  Both traditional and Web-based operations can help customers quickly locate products of interest to them and simultaneously increase the value of each communication with the customers.
- *Learn parameters influencing trends in sales and margins.*
  In the majority of cases we have no clue on what combination of parameters influences operation (black box). In these situations data mining is the only real option.
- *Segment markets and personalize communications.*
  There might be distinct groups of customers, patients, or natural phenomena that require different approaches in their handling.

The importance of collecting data that reflect specific business or scientific activities to achieve competitive advantage is widely recognized. Powerful systems for collecting data and managing it in large databases are in place in all large and mid-range companies. However, the bottleneck of turning this data into information is the difficulty of extracting knowledge about the system being studied from the collected data. Human analysts without special tools can no longer make sense of enormous volumes of data that require processing in order to make informed business decisions (Kudyba & Hoptroff, 2001).

The applications of data mining are everywhere: from biomedical data (Hu and Xu, 2005) to mobile user data (Goh and Taniar, 2005); from data warehousing (Tjioe and Taniar, 2005) to intelligent web personalization (Zhou, Cheung, & Fong, 2005); from analyzing clinical outcome (Hu, Song, Han, Yoo, Prestrud, Brennan, & Brooks, 2005) to mining crime patterns (Bagui, 2006).

Potential Pitfalls

Data Quality

Data quality means the accuracy and completeness of the data. Data quality is a versatile issue that represents one of the biggest challenges for data mining.

Data quality problem is of great importance due to the emergence of large volumes of data. Many business and industrial applications critically rely on the quality of information stored in diverse databases and data warehouses. As Seifert (2004) emphasized that data quality can be affected by the structure and consistency of the data being analyzed. Other factors like the presence of duplicate records, the lack of data standards, the timeliness of updates and human errors can significantly impact the effectiveness of complex data mining techniques, which are sensitive to subtle differences in data. To improve the quality of data it is sometimes necessary to clean data by removing the duplicate records, standardizing the values or symbols used in the database to represent certain information, accounting for missing data points, removing unneeded data fields, identifying abnormal data points.

Interoperability

Interoperability refers to the ability of computer system and/or data to work with other systems or data using common standards or process. Until recently, some government agencies elected not to gamble with any level of open access and operated isolated information systems. But isolated data is in many ways useless data; bits of valuable information on the Sept. 11, 2001 hijackers' activities may have been stored in a variety of databases at the federal, state, and local government levels, but that information was not colleted and available to those who needed to see it to glimpse a complete picture of the growing threat. So Seifert (2004) suggested that it is a critical part of the larger efforts to improve interagency collaboration and information sharing. For public data mining, interoperability of databases and software is important to enable the search and analysis of multiple databases simultaneously. This also ensures the compatibility of data mining activities of different agencies.

Standardization

This allows you to arrange customer information in a consistent format. Among the biggest challenges are inconsistent abbreviations, and misspellings and variant spellings. Among the types of data that can be appended are demographic, geographic, psychographic, behavioristic, event-driven and computed. Matching allows you to identify similar data within and across your data sources. One of the greatest challenges of matching is creating a system that incorporates your "business rules," or criteria for determining what constitutes a match.

Preventing Decay

The worst enemy of information is time. And information decays at different rates (Berry & Linoff, 2004). Cleaning your database is a large accomplishment, but it will be short-lived if you fail to implement procedures for keeping it clean at the source. According to the second law of thermodynamics, ordered systems tend to disorder, and a database is a very ordered system. Contacts move. Companies grow. Knowledge workers enter new customer information incorrectly.

Some information simply starts out wrong, result of data input errors such as typos, transpositions, omissions and other mistakes. These are often easy to avoid. Finding ways to successfully implement these new technologies into a comprehensive data quality program not only increases the quality of your customer information, but also saves time, reduces frustration, improves customer relations, and ultimately increases revenue. Without constant attention to quality, your information quality will disintegrate.

No Generalizations to a Population

In statistics a population is defined, and then a sample is collected to make inferences about the population. This means that data cannot be re-used. They define a model before looking at the data. Data mining does not attempt generalizations to a population. The database is considered as the population. With the computing power of modern computers data miners can use the whole database, making sampling redundant. Data can be re-used. In data mining it is a common practice to try hundreds of models and find the one that fits best. This makes the interpretation of the significance difficult. Machine learning is the data mining equivalent to regression. In machine learning we use a training set to train the system to find the dependent variable.

FUTURE TRENDS

Predictive Analysis

Augusta (2004) suggested that predictive analysis is one of the major future trends for data mining. Rather than being just about mining large amounts of data, predictive analytics looks to actually understand the data content. They hope to forecast based on the contents of the data. However this requires complex programming and a great amount of business acumen. They are looking to do more than simply archive data, which is what data mining is currently known for. They want to not just process it, but understand it more clearly which will in turn allow them to make better predictions about future behavior. With predictive analytics you have the program scour the data and try to form, or help form, new hypotheses itself. This shows great promise, and would be a boon for industries everywhere.

Diversity of Application Domains

Data mining and X" phenomenon, as Tuzhilin (2006) coined, where X constitutes a broad range of fields in which data mining is used for analyzing the data. This has resulted in a process of cross-fertilization of ideas generated within this diverse population of researchers interacting across the traditional boundaries of their disciplines. The next generation of data mining applications covers a large number of different fields from traditional businesses to advance scientific research. Kantardzic & Zurada (2005) observed that with new tools, methodologies, and infrastructure, this trend of diversification will continue each year.

CONCLUSION

The emergence of new information technologies has given us much more data and many more options how to use it. Yet managing that flood of data, and making it useful and available to decision makers has been a major organizational challenge. Data mining allows the extraction of diamonds of knowledge from huge historical mines of data. It helps to predict outcomes of future situations, to optimize business decisions, to increase the value of each customer and communication, and to improve customer satisfaction.

The management of data requires understanding and a skill set far beyond mere programming. Managing data mining is a new revelation as analysts will have to sift through more and more information daily due to the ever increasing size of the Web and consumer purchases. Data mining can have enormous rewards if properly used. We have an unprecedented opportunity for the future is we could avoid data mining's pitfalls.

REFERENCES

Augusta, L. (2004, August). The future of data mining - predictive analytics. *DM Review*, 14 (8), 16-20, 37.

Bagui, S. (2006). An approach to mining crime patterns, *International Journal of Data Warehousing and Mining*, Idea Group Inc., 2(1), 50-80.

Berry, M. J. A, & Linoff, G. S. (2004). Data mining techniques: for marketing, sales, and customer relationship management. *Wiley Publishers.*

Davenport, T., & Harris, J. G. (2007). *Competing on analytics: The new science of winning.* Harvard Business School Press.

Edelstein, H., & Millenson, J. (2003, December). Data mining in depth: data mining and privacy. *DM Review.* Retrieved September 27, 2007, from http://www.dmreview.com/editorial/dmreview/print_action.cfm?articleId=7768

Goh, J., and Taniar, D. (2005). Mining parallel patterns from mobile users, *International Journal of Business Data Communications and Networking,* 1(1), 50-76.

Han J, and Kamber, M. (2001). *Data mining: concepts and techniques.* San Francisco, CA: Morgan Kaufmann Publishers.

Hand, D. J., Mannila, H. & Smyth, P. (2001, August). *Principles of data mining.* The MIT Press.

Hu, X., Song, I-Y., Han, H., Yoo, I., Prestrud, A. A., Brennan, M.F., & Brooks, A. D. (2005). Temporal rule induction for clinical outcome analysis, *International Journal of Business Intelligence and Data Mining*, Inderscience Publishers, 1(1), 122-136.

Hu, X., Xu, X (2005). Mining novel connections from online biomedical text databases using semantic query expansion and semantic-relationship pruning, *Interna-*

D

tional Journal of Web and Grid Services, Inderscience Publishers, 1(2), 222-239.

Kantardzic, M. M., & Zurada, J. (Ed.) (2005). *Next generation of data-mining applications*, IEEE Press, Wiley-Interscience.

Kudyba, S., & Hoptroff, R. (2001). Data mining and business intelligence: a guide to productivity. *Ideal Group Publishing.*

Lavoie, B., Dempsey, L., & Connaway, L. S. (2006). Making data work harder. *Library Journal*, 131 (1), 40-43.

Megaputer. (2006). Data mining. Retrieved Sept. 27, 2007, from http://www.megaputer.com/dm/

Olafsson, S. (2006). Introduction to operations research and data mining. *Computers & Operations Research*, 33 (11), 3067-3069.

Seiftert, J. (2004, December 16). Data mining: an overview. *CRS Report for Congress*. Retrieved September 27, 2007, from http://www.fas.org/irp/crs/RL31798.pdf

Tjioe, H.C. and Taniar, D. (2005). Mining association rules in data warehouses, *International Journal of Data Warehousing and Mining*, 1(3), 28-62.

Tuzhilin, A. (2006). Foreword. In Wang, J. (Ed.). *Encyclopedia of data warehousing and mining* (2 Volumes), First Edition. Hershey, PA: Idea Group Reference.

Wang, J. (Ed.) (2006). *Encyclopedia of data warehousing and mining* (2 Volumes), First Edition. Hershey, PA: Idea Group Reference.

Wang, J. (2003). *Data mining: opportunities and challenges*, Hershey, PA: Idea Group Publishing.

Zhou, B., Cheung, S., and Fong, A. C. M. (2005). A Web usage lattice based mining approach for intelligent Web personalization, *International Journal of Web Information Systems*, Troubador Publishing, UK, 1(3).

Zhou, Z. (2003). Three perspectives of data mining. *Artificial Intelligence*, 14, 139-146.

KEY TERMS

Data Mining: The process of automatically searching large volumes of data for patterns. Data mining is a fairly recent and contemporary topic in computing.

Data Visualization: A technology for helping users to see patterns and relationships in large amounts of data by presenting the data in graphical form.

Explanatory Variables: Used interchangeably and refer to those variables that explain the variation of a particular target variable. Also called driving, or descriptive, or independent variables.

Information Quality Decay: Quality of some data goes down when facts about real world objects change over time, but those facts are not updated in the database.

Information Retrieval: The art and science of searching for information in documents, searching for documents themselves, searching for metadata which describe documents, or searching within databases, whether relational stand-alone databases or hypertext networked databases such as the Internet or intranets, for text, sound, images or data.

Machine Learning: Concerned with the development of algorithms and techniques, which allow computers to "learn".

Neural Networks: Also referred to as artificial intelligence (AI), which utilizes predictive algorithms.

Pattern Recognition: The act of taking in raw data and taking an action based on the category of the data. It is a field within the area of machine learning.

Predictive Analysis: Use of data mining techniques, historical data, and assumptions about future conditions to predict outcomes of events.

Segmentation: Another major group that comprises the world of data mining involving technology that identifies not only statistically significant relationships between explanatory and target variables, but determines noteworthy segments within variable categories that illustrate prevalent impacts on the target variable.

Data Warehousing Development and Design Methodologies

James Yao
Montclair State University, USA

John Wang
Montclair State University, USA

INTRODUCTION

Information systems were developed in early 1960s to process orders, billings, inventory controls, payrolls, and accounts payables. Soon information systems research began. Harry Stern started the "Information Systems in Management Science" column in Management Science journal to provide a forum for discussion beyond just research papers (Banker & Kauffman, 2004). Ackoff (1967) led the earliest research on management information systems for decision-making purposes and published it in *Management Science*. Gorry and Scott Morton (1971) first used the term 'decision support systems' (DSS) in a paper and constructed a framework for improving management information systems. The topics on information systems and DSS research diversifies. One of the major topics has been on how to get systems design right.

As an active component of DSS, which is part of today's business intelligence systems, data warehousing became one of the most important developments in the information systems field during the mid-to-late 1990s. Since business environment has become more global, competitive, complex, and volatile, customer relationship management (CRM) and e-commerce initiatives are creating requirements for large, integrated data repositories and advanced analytical capabilities. By using a data warehouse, companies can make decisions about customer-specific strategies such as customer profiling, customer segmentation, and cross-selling analysis (Cunningham et al., 2006). Thus how to design and develop a data warehouse have become important issues for information systems designers and developers.

This paper presents some of the currently discussed development and design methodologies in data warehousing, such as the multidimensional model vs. relational ER model, CIF vs. multidimensional meth- odologies, data-driven vs. metric-driven approaches, top-down vs. bottom-up design approaches, data partitioning and parallel processing.

BACKGROUND

Data warehouse design is a lengthy, time-consuming, and costly process. Any wrongly calculated step can lead to a failure. Therefore, researchers have placed important efforts to the study of design and development related issues and methodologies.

Data modeling for a data warehouse is different from operational database data modeling. An operational system, e.g., online transaction processing (OLTP), is a system that is used to run a business in real time, based on current data. An OLTP system usually adopts Entity-relationship (ER) modeling and application-oriented database design (Han & Kamber, 2006). An information system, like a data warehouse, is designed to support decision making based on historical point-in-time and prediction data for complex queries or data mining applications (Hoffer, et al., 2007). A data warehouse schema is viewed as a dimensional model (Ahmad et al., 2004, Han & Kamber, 2006; Levene & Loizou, 2003). It typically adopts either a star or snowflake schema and a subject-oriented database design (Han & Kamber, 2006). The schema design is the most critical to the design of a data warehouse.

Many approaches and methodologies have been proposed in the design and development of data warehouses. Two major data warehouse design methodologies have been paid more attention. Inmon et al. (2000) proposed the Corporate Information Factory (CIF) architecture. This architecture, in the design of the atomic-level data marts, uses denormalized entity-relationship diagram (ERD) schema. Kimball (1996, 1997) proposed multidimensional (MD) architecture.

This architecture uses star schema at atomic-level data marts. Which architecture should an enterprise follow? Is one better than the other?

Currently, the most popular data model for data warehouse design is the dimensional model (Han & Kamber, 2006; Bellatreche, 2006). Some researchers call this model the data-driven design model. Artz (2006), nevertheless, advocates the metric-driven model, which, as another view of data warehouse design, begins by identifying key business processes that need to be measured and tracked over time in order for the organization to function more efficiently. There has always been the issue of top-down vs. bottom-up approaches in the design of information systems. The same is with a data warehouse design. These have been puzzling questions for business intelligent architects and data warehouse designers and developers. The next section will extend the discussion on issues related to data warehouse design and development methodologies.

DESIGN AND DEVELOPMENT METHODOLOGIES

Data Warehouse Data Modeling

Database design is typically divided into a four-stage process (Raisinghani, 2000). After requirements are collected, conceptual design, logical design, and physical design follow. Of the four stages, logical design is the key focal point of the database design process and most critical to the design of a database. In terms of an OLTP system design, it usually adopts an ER data model and an application-oriented database design (Han & Kamber, 2006). The majority of modern enterprise information systems are built using the ER model (Raisinghani, 2000). The ER data model is commonly used in relational database design, where a database schema consists of a set of entities and the relationship between them. The ER model is used to demonstrate detailed relationships between the data elements. It focuses on removing redundancy of data elements in the database. The schema is a database design containing the logic and showing relationships between the data organized in different relations (Ahmad et al., 2004). Conversely, a data warehouse requires a concise, subject-oriented schema that facilitates online data analysis. A data warehouse schema is viewed as a dimensional model which is composed of a central fact table and a set of surrounding dimension tables, each corresponding to one of the components or dimensions of the fact table (Levene & Loizou, 2003). Dimensional models are oriented toward a specific business process or subject. This approach keeps the data elements associated with the business process only one join away. The most popular data model for a data warehouse is multidimensional model. Such a model can exist in the form of a star schema, a snowflake schema, or a starflake schema.

The star schema (see Figure 1) is the simplest database structure containing a fact table in the center, no redundancy, which is surrounded by a set of smaller dimension tables (Ahmad et al., 2004; Han & Kamber, 2006). The fact table is connected with the dimension tables using many-to-one relationships to ensure their hierarchy. The star schema can provide fast response time allowing database optimizers to work with simple database structures in order to yield better execution plans.

The snowflake schema (see Figure 2) is a variation of the star schema model, in which all dimensional information is stored in the third normal form, thereby further splitting the data into additional tables, while keeping fact table structure the same. To take care of hierarchy, the dimension tables are connected with sub-dimension tables using many-to-one relationships. The resulting schema graph forms a shape similar to a snowflake (Ahmad et al., 2004; Han & Kamber, 2006). The snowflake schema can reduce redundancy and save storage space. However, it can also reduce the effectiveness of browsing and the system performance may be adversely impacted. Hence, the snowflake schema is not as popular as star schema in data warehouse design (Han & Kamber, 2006). In general, the star schema requires greater storage, but it is faster to process than the snowflake schema (Kroenke, 2004).

The starflake schema (Ahmad et al., 2004), also called galaxy schema or fact constellation schema (Han & Kamber, 2006), is a combination of the denormalized star schema and the normalized snowflake schema (see Figure 3). The starflake schema is used in situations where it is difficult to restructure all entities into a set of distinct dimensions. It allows a degree of crossover between dimensions to answer distinct queries (Ahmad et al., 2004). Figure 3 illustrates the starflake schema.

What needs to be differentiated is that the three schemas are normally adopted according to the differ-

D

Figure 1. Example of a star schema (adapted from Kroenke, 2004)
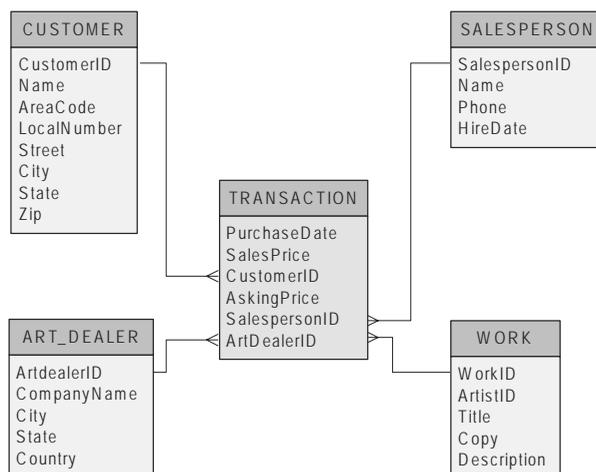
Figure 2. Example of a snowflake schema (adapted from Kroenke, 2004)
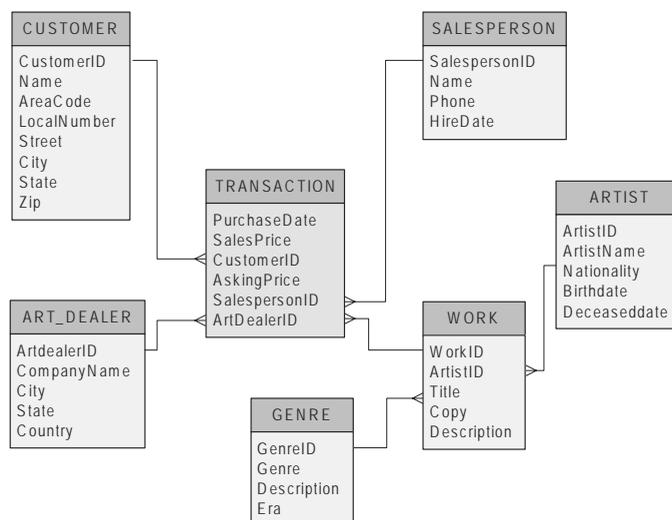
ences of design requirements. A data warehouse collects information about subjects that span the entire organization, such as customers, items, sales, etc. Its scope is enterprise-wide (Han & Kamber, 2006). Starflake schema can model multiple and interrelated subjects. Therefore, it is usually used to model an enterprise-wide data warehouse. A data mart, on the other hand, is similar to a data warehouse but limits its focus to a department subject of the data warehouse. Its scope is department-wide. The star schema and snowflake schema are geared towards modeling single subjects. Consequently, the star schema or snowflake schema is commonly used for a data mart modeling, although the star schema is more popular and efficient (Han & Kamber, 2006).

Figure 3. Example of a starflake schema (galaxy schema or fact constellation) (adapted from Han & Kamber, 2006)

D

TIME	SALES	ITEM	SHIPPING	SHIPPER
Time_key	Time_key	Item_key	Item_key	Shipper_key
Day	Item_key	Item_name	Time_key	Shipper_name
Day_of_week	Branch_key	Brand	Shipper_key	Location_key
Month	Location_key	Type	From_location	Shipper_type
Quarter	Dollars_sold	Supplier_type	To_location	
Year	Units_sold		Dollars_cost	
			Units_shipped	

BRANCH	LOCATION
Branch_key	Location_key
Branch_name	Street
Branch_type	City
	Province_or_state
	Country

CIF vs. Multidimensional

Two major design methodologies have been paid more attention in the design and development of data warehouses. Kimball (1996, 1997) proposed multidimensional (MD) architecture. Inmon, Galemmco, and Geiger (2000) proposed the Corporate Information Factory (CIF) architecture. Imhoff et al. (2004) made a comparison between the two by using important criteria, such as scope, perspective, data flow, etc. One of the most significant differences between the CIF and MD architectures is the definition of data mart. For MD architecture, the design of the atomic-level data marts is significantly different from the design of the CIF data warehouse, while its aggregated data mart schema is approximately the same as the data mart in the CIF architecture. MD architecture uses star schemas, whereas CIF architecture uses denormalized ERD schema. This data modeling difference constitutes the main design difference in the two architectures (Imhoff et al., 2004). A data warehouse may need both types of data marts in the data warehouse bus architecture depending on the business requirements. Unlike the CIF architecture, there is no physical repository equivalent to the data warehouse in the MD architecture.

The design of the two data marts is predominately multidimensional for both architecture, but the CIF architecture is not limited to just this design and can support a much broader set of data mart design techniques. In terms of scope, both architectures deal with enterprise scope and business unit scope, with CIF architecture putting a higher priority on enterprise scope and MD architecture placing a higher priority on business unit scope. Imhoff et al. (2004) encourage the application of a combination of the data modeling techniques in the two architectural approaches, namely, the ERD or normalization techniques for the data warehouse and the star schema data model for multidimensional data marts. A CIF architecture with only a data warehouse and no multidimensional marts is almost useless and a multidimensional data-mart-only environment risks the lack of an enterprise integration and support for other forms of business intelligence analyses.

Data-Driven vs. Metric-Driven

Currently, the most popular data model for data warehouse design is the dimensional model (Han & Kamber, 2006; Bellatreche, 2006). In this model, data from OLTP systems are collected to populated dimensional model. Researchers term a data warehouse design based on this model as a data-driven design model since the information acquisition processes in the data warehouse are driven by the data made available in the underlying operational information systems. Another view of data warehouse design is called the metric-driven view (Artz, 2006), which begins by identifying key business processes that need to be measured and tracked over time in order for the organization to function more efficiently. Advantages of data-driven model include that it is more concrete, evolutionary, and uses derived summary data. Yet the information generated from the data warehouse may be meaningless to the user owing to the fact that the nature of the

derived summary data from OLTP systems may not be clear. The metric-driven design approach, on the other hand, begins first by defining key business processes that need to be measured and tracked over time. After these key business processed are identified, then they are modeled in a dimensional data model. Further analysis follows to determine how the dimensional model will be populated (Artz, 2006).

According to Artz (2006), data-driven model to a data warehouse design has little future since information derived from a data-driven model is information about the data set. Metric-driven model, conversely, is possibly to have some key impacts and implications because information derived from a metric-driven model is information about the organization. Data-driven approach is dominating data warehouse design in organizations at present. Metric-driven, on the other hand, is at its research stage, needing practical application testimony of its speculated potentially dramatic implications.

Top-Down vs. Bottom-Up

There are two approaches in general to building a data warehouse prior to the data warehouse construction commencement, including data marts: the top-down approach and bottom-up approach (Han & Kamber, 2006; Imhoff et al., 2004; Marakas, 2003). Top-down approach starts with a big picture of the overall, enterprise-wide design. The data warehouse to be built is large and integrated, with a focus on integrating the enterprise data for usage in any data mart from the very first project (Imhoff et al., 2004). It implies a strategic rather than an operational perspective of the data. It serves as the proper alignment of an organization's information systems with its business goals and objectives (Marakas, 2003). However, this approach is risky (Ponniah, 2001). In contrast, a bottom-up approach is to design the warehouse with business-unit needs for operational systems. It starts with experiments and prototypes (Han & Kamber, 2006). With bottom-up, departmental data marts are built first one by one. It offers faster and easier implementation, favorable return on investment, and less risk of failure, but with a drawback of data fragmentation and redundancy. The focus of bottom-up approach is to meet unit-specific needs with minimum regards to the overall enterprise-wide data requirements (Imhoff et al, 2004).

An alternative to the above-discussed two approaches is to use a combined approached (Han & Kamber, 2006), with which "an organization can exploit the planned and strategic nature of the top-down approach while retaining the rapid implementation and opportunistic application of the bottom-up approach" (p. 129), when such an approach is necessitated in the undergoing organizational and business scenarios.

Data Partitioning and Parallel Processing

Data partitioning is the process of decomposing large tables (fact tables, materialized views, indexes) into multiple small tables by applying the selection operators (Bellatreche, 2006). A good partitioning scheme is an essential part of designing a database that will benefit from parallelism (Singh, 1998). With a well performed partitioning, significant improvements in availability, administration, and table scan performance can be achieved.

Parallel processing is based on a parallel database, in which multiprocessors are in place. Parallel databases link multiple smaller machines to achieve the same throughput as a single, larger machine, often with greater scalability and reliability than single processor databases (Singh, 1998). In a context of relational online analytical processing (ROLAP), by partitioning data of ROLAP schema (star schema or snowflake schema) among a set of processors, OLAP queries can be executed in a parallel, potentially achieving a linear speedup and thus significantly improving query response time (Datta et al., 1998; Tan, 2006). Given the size of contemporary data warehousing repositories, multiprocessor solutions are crucial for the massive computational demands for current and future OLAP system (Dehne et al., 2006). The assumption of most of the fast computation algorithms is that their algorithms can be applied into the parallel processing system (Dehne, 2006; Tan, 2006). As a result, it is sometimes necessary to use parallel processing for data mining because large amounts of data and massive search efforts are involved in data mining (Turban et al., 2005). Therefore, data partitioning and parallel processing are two complementary techniques to achieve the reduction of query processing cost in data warehousing design and development (Bellatreche, 2006).

FUTURE TRENDS

Currently, data warehousing is largely applied in customer relationship management (CRM). However, there are up to date no agreed upon standardized rules for how to design a data warehouse to support CRM and a taxonomy of CRM analyses needs to be developed to determine factors that affect design decisions for CRM data warehouse (Cunningham et al., 2006).

In data modeling area, to develop a more general solution for modeling data warehouse current ER model and dimensional model need to be extended to the next level to combine the simplicity of the dimensional model and the efficiency of the ER model with the support of object oriented concepts.

CONCLUSION

Several data warehousing development and design methodologies have been reviewed and discussed. Data warehouse data model differentiates itself from ER model with an orientation toward specific business purposes. It benefits an enterprise greater if the CIF and MD architectures are both considered in the design of a data warehouse. Some of the methodologies have been practiced in the real world and accepted by today's businesses. Yet new challenging methodologies, particularly in data modeling and models for physical data warehousing design, such as the metric-driven methodology, need to be further researched and developed.

REFERENCES

Ackoff, R.I. (1967). Management misinformation systems. *Management Science*, 14(4), 147-156.

Ahmad, I., Azhar, S., & Lukauskis, P. (2004). Development of a decision support system using data warehousing to assist builders/developers in site selection. *Automation in Construction*, 13, 525-542.

Artz, J. M. (2006). Data driven vs. metric driven data warehouse design. In J. Wang (Ed.), *Encyclopedia of Data Warehousing and Mining* (Vol. 1, pp. 223-227). Hershey, PA: Idea Group References.

Banker, R.D., & Kauffman, R.J. (2004, March). The evolution of research on information systems: A fiftieth-year survey of the literature in *Management Science*. *Management Science*, 50(3), 281-298.

Bellatreche, L., & Mohania, M. (2006). Physical data warehousing design. In J. Wang (Ed.), *Encyclopedia of Data Warehousing and Mining* (Vol. 2, pp. 906-911). Hershey, PA: Idea Group References.

Bellatreche, L., Schneider, M., Mohania, M., & Bhargava, B. (2002). PartJoin: An efficient storage and query execution for data warehouses. *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery (DAWAK'02)*, 109-132.

Cunningham, C., Song, I., & Chen, P.P. (2006, April-June). Data warehouse design to support customer relationship management analyses. *Journal of Database Management*, 17(2), 62-84.

Datta, A., Moon, B., & Thomas, H. (1998). A case for parallelism in data warehousing and OLAP. *Proceedings of the 9th International Workshop on Database and Expert Systems Applications (DEXA'98)*, 226-231.

Dehne, F., Eavis, T., & Rau-Chaplin, A. (2006). The cgmCUBE project: Optimizing parallel data cube generation for ROLAP. *Distrib. Parallel Databases*, 19, 29-62.

Gorry, G.A., & Scott Morton, M.S. (1971). A framework for management information systems. *Sloan Management Review*, 13(1), 1-22.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques* (2nd ed.). San Francisco, CA: Morgan Kaufmann Publisher.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge Massachusetts: The MIT Press.

Hoffer, J.A., Prescott, M.B., & McFadden, F.R. (2007). *Modern database management* (8th ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Imhoff, C., Galemmco, M., & Geiger, J.G. (2004). Comparing two data warehouse methodologies. (Database and network intelligence). *Database and Network Journal*, 34(3), 3-9.

Inmon, W.H., Terdeman, R.H., & Imhoff, C. (2000). *Exploration warehousing: Turning business informa-*

D

tion into business opportunity. NY: John Wiley & Sons, Inc.

Kimball, R. (1996). *The data warehouse toolkit: Practical techniques for building dimensional data warehouses*. NY: John Wiley & Sons.

Kimball, R. (1997, August). A dimensional modeling manifesto. *DBMS*, 10(9), 58-70.

Kroenke, D.M. (2004). *Database processing: Fundamentals, design and implementation* (9th ed.). Saddle River, New Jersey: Prentice Hall.

Levene, M., & Loizou, G. (2003). Why is the snowflake schema a good data warehouse design? *Information Systems*, 28(3), 225-240.

Marakas, G.M. (2003). *Modern data warehousing, mining, and visualization: Core concepts*. Upper Saddle River, New Jersey: Pearson Education Inc.

Ponniah, P. (2001). *Data warehousing fundamentals: A comprehensive guide for IT professionals*. New York: John Wiley & Sons, Inc.

Raisinghani, M.S. (2000). Adapting data modeling techniques for data warehouse design. *Journal of Computer Information Systems*, 4(3), 73-77.

Singh, H.S. (1998). *Data warehousing: Concepts, technologies, implementations, and management*. Upper Saddle River, NJ: Prentice Hall PTR.S

Tan, R.B. (2006). Online analytical processing systems. In J. Wang (Ed.), *Encyclopedia of Data Warehousing and Mining* (Vol. 2, pp. 876-884). Hershey, PA: Idea Group References.

KEY TERMS

Dimensions: They are the perspectives or entities with respect to which an organization wants to keep records (Han & Kamber, 2006, p. 110).

Dimensional Model: A model containing a central fact table and a set of surrounding dimension tables, each corresponding to one of the components or dimensions of the fact table.

Entity-Relationship Data Model: A model that represents database schema as a set of entities and the relationships among them.

Fact Table: The central table in a star schema, containing the names of the facts, or measures, as well as keys to each of the related dimension tables.

Metric-Drive Design: A data warehousing design approach which begins by defining key business processes that need to be measured and tracked over time. Then they are modeled in a dimensional model.

Parallel Processing: The allocation of the operating system's processing load across several processors (Singh, 1998, p. 209).

Star Schema: A modeling diagram which contains a large central table (fact table) and a set of smaller attendant tables (dimension tables) each represented by only one table with a set of attributes.

Decision Making in Intelligent Agents

Mats Danielson
Stockholm University, Sweden & Royal Institute of Technology, Sweden

Love Ekenberg
Stockholm University, Sweden & Royal Institute of Technology, Sweden

INTRODUCTION

There are several ways of building complex distributed software systems, for example in the form of software agents. But regardless of the form, there are some common problems having to do with specification contra execution. One of the problems is the inherent dynamics in the environment many systems are exposed to. The properties of the environment are not known with any precision at the time of construction. This renders a specification of the system incomplete by definition. A traditional **software agent** is only prepared to handle situations conceived of and implemented at compile-time. Even though it can operate in varying contexts, its decision making abilities are static. One remedy is to prepare the distributed components for a truly dynamic environment, i.e. an environment with changing and somewhat unpredictable conditions. A rational **software agent** needs both a representation of a decision problem at hand and means for evaluation. AI has traditionally addressed some parts of this problem such as representation and reasoning, but has hitherto to a lesser degree addressed the decision making abilities of independent distributed software components (Ekenberg, 2000a, 2000b). Such decision making often has to be carried out under severe uncertainty regarding several parameters. Thus, methods for independent **decision making** components should be able to handle uncertainties on the probabilities and utilities involved. They have mostly been studied as means of representation, but are now being developed into functional theories of decision making suitable for dynamic use by software agents and other dynamic distributed components. Such a functional theory will also benefit analytical decision support systems intended to aid humans in their decision making. Thus, the generic term *agent* below stands for a dynamic software component as well as a human or a group of humans assisted by intelligent software.

BACKGROUND

Ramsey (1926/78) was the first to suggest a theory that integrated ideas on subjective probability and utility in presenting (informally) a general set of axioms for preference comparisons between acts with uncertain outcomes (probabilistic decisions). von Neumann and Morgenstern (1947) established the foundations for a modern theory of utility. They stated a set of axioms that they deemed reasonable to a rational decision-maker (such as an agent), and demonstrated that the agent should prefer the alternative with the highest expected utility, given that she acted in accordance with the axioms. This is the principle of maximizing the **expected utility**. Savage (1954/72) published a thorough treatment of a complete theory of subjective expected utility. Savage, von Neumann, and others structured decision analysis by proposing reasonable principles governing decisions and by constructing a theory out of them. In other words, they (and later many others) formulated a set of axioms meant to justify their particular attitude towards the utility principle, cf., e.g., Herstein and Milnor (1953), Suppes (1956), Jeffrey (1965/83), and Luce and Krantz (1971). In classical **decision analysis**, of the types suggested by Savage and others, a widespread opinion is that utility theory captures the concept of rationality.

After Raiffa (1968), probabilistic decision models are nowadays often given a tree representation (see Fig. 1). A **decision tree** consists of a root, representing a decision, a set of event nodes, representing some kind of uncertainty and consequence nodes, representing possible final outcomes. In the figure, the decision is a square, the events are circles, and final consequences are triangles. Events unfold from left to right, until final consequences are reached. There may also be more than one decision to make, in which case the sub-decisions are made before the main decision.

Figure 1. Decision tree
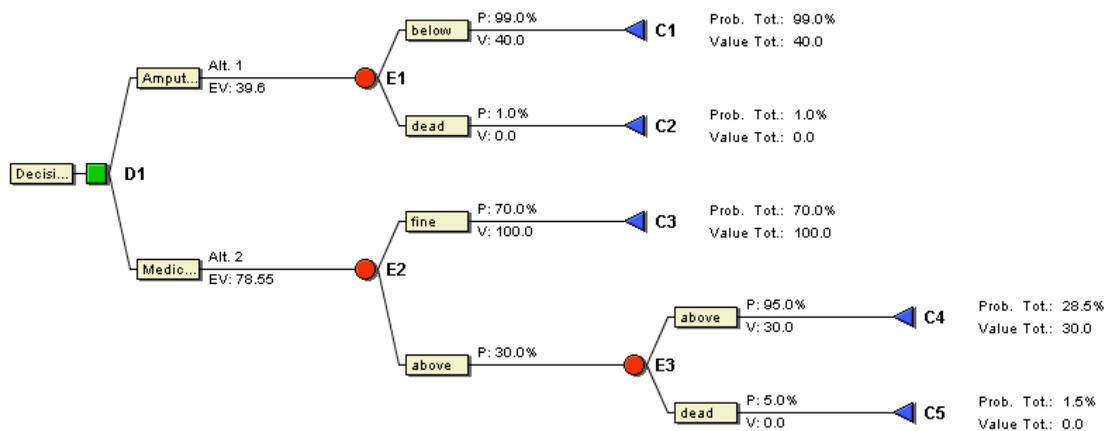
In decision trees, probability distributions are assigned in the form of weights (numbers) in the probability nodes as measures of the uncertainties involved. Obviously, such a numerically precise approach puts heavy demands on the input capability of the agent. The shortcomings of this representation are many, and have to be compensated for, see, e.g., (Ekenberg, 2000a). Among other things, the question has been raised whether people are capable of providing the input information that utility theory requires (cf., e.g., (Fischhoff et al., 1983)). For instance, most people cannot clearly distinguish between probabilities ranging roughly from 0.3 to 0.7 (Shapira, 1995). Similar problems arise in the case of artificial agents, since utility-based artificial agents usually base their reasoning on human assessments, for instance in the form of induced preference functions. The so-called reactive agents, for which this does not hold true, have not been put to use in dynamic domains involving uncertainty (cf., e.g., (Russell & Norvig, 1995)). Furthermore, even if an agent would be able to discriminate between different probabilities, very often complete, adequate, and precise information is missing.

Consequently, during recent years of rather intense research activities several alternative approaches have emerged. In particular, first-order approaches, i.e., based on sets of probability measures, upper and lower probabilities, and **interval probabilities**, have prevailed. A main class of such models has been focused on expressing probabilities in terms of intervals. In 1953, the concept of capacities was introduced (Choquet, 1953/54). This representation approach was further

developed in (Huber, 1973, Huber & Strassen, 1973). Capacities have subsequently been used for modelling imprecise probabilities as intervals (capacities of order 2 (Denneberg, 1994)). Since the beginning of the 1960s the use of **first-order** (interval-valued) probability functions, by means of classes of probability measures, has been integrated in classical probability theory by, e.g., Smith (1961) and Good (1962). Similarly, Dempster (1967) investigated a framework for modelling upper and lower probabilities, which was further developed by Shafer (1976), where a representation of belief in states or events was provided. Within the AI community the Dempster-Shafer approach has received a good deal of attention. However, their formalism seems to be too strong to be an adequate representation of belief (Weichselberger & Pöhlman, 1990).

Other representations in terms of upper and lower probabilities have been proposed by, i.a., Hodges and Lehmann (1952), Hurwicz (1951), Wald (1950), Kyburg (1961), Levi (1974, 1980), Walley (1991), Danielson and Ekenberg (1998, 2007), and Ekenberg et al. (2001). Upper and lower previsions have also been investigated by various authors. For instance, Shafer et al. (2003) suggests a theory for how to understand subjective probability estimates based on Walley (1991). A few approaches have also been based on logic, e.g., Nilsson (1986). He develops methods for dealing with sentences involving upper and lower probabilities. This kind of approaches has been pursued further by, among others, Wilson (1999).

SECOND-ORDER REPRESENTATIONS

A common characteristic of the first-order representations above is that they typically do not include all of the strong axioms of probability theory and thus they do not require an agent to model and evaluate a decision situation using precise probability (and, in some cases, value) estimates. An advantage of representations using upper and lower probabilities is that they do not require taking probability distributions into consideration. On the other hand, it is then often difficult to devise a reasonable decision rule that finds an admissible alternative out of a set of alternatives and at the same time fully reflects the intensions of an agent (or its owner). Since the probabilities and values are represented by intervals, the expected value range of an alternative will also be an interval. In effect, the procedure retains all alternatives with overlapping expected utility intervals, even if the overlap is very small. Furthermore, they do not admit for discrimination between different beliefs in different values within the intervals.

All of these representations face the same trade-off. **Zero-order** approaches (i.e. fixed numbers representing probability and utility assessments) require unreasonable precision in the representation of input data. Even though the evaluation and discrimination between alternatives becomes simple, the results are often not a good representative of the problem and sensitivity analyses are hard to carry out for more than a few parameters at a time. **First-order** approaches (e.g. intervals) offer a remedy to the representation problem by allowing imprecision in the representation of probability and utility assessments such as intervals, reflecting the uncertainty inherent in most real-life decision problems faced by agents. But this permissibility opens up a can of worms in the sense that evaluation and discrimination becomes much harder because of overlap in the evaluation results of different options for the agent, i.e. the worst case for one alternative is no better than the best case for another alternative or vice versa, rendering a total ranking order between the alternatives impossible to achieve. The trade-off between realistic representation and discriminative power has not been solved within the above paradigms. For a solution, one must look at **second-order** approaches allowing both imprecision in representation and power of admissible discrimination.

Approaches for extending the interval representation using distributions over classes of probability and value measures have been developed into various hierarchical models, such as second-order probability theory (Gärdenfors & Sahlin, 1982, 1983, Ekenberg & Thorbiörnson, 2001, Ekenberg et al., 2005). Gärdenfors and Sahlin consider global distributions of beliefs, but restrict themselves to interval representations and only to probabilities, not utilities. Other limitations are that they neither investigate the relation between global and local distributions, nor do they introduce methods for determining the consistency of user-asserted sentences. The same applies to Hodges and Lehmann (1952), Hurwicz (1951), and Wald (1950). Some more specialized approaches have recently been suggested, such as (Jaffray, 1999), (Nau, 2002), and (Utkin & Augustin, 2003). In general, very few have addressed the problems of computational complexity when solving decision problems involving such estimates. Needless to say, it is important in dynamic agents to be able to determine, in a reasonably short time, how various evaluative principles rank the given options in a decision situation.

Ekenberg et al. (2006) and Danielson et al. (2007) provide a framework for how second-order representation can be systematically utilized to put belief information into use in order to efficiently discriminate between alternatives that evaluate into overlapping expected utility intervals when using first-order interval evaluations. The belief information is in the form of a joint belief distribution, specified as marginal belief distributions projected on each parameter. It is shown that regardless of the form of belief distributions over the originating intervals, the distributions resulting from multiplications and additions have forms very different from their components. This warp of resulting belief demonstrates that analyses using only first-order information such as upper and lower bounds are not taking all available information into account. The method is based on the agent's belief in different parts of the intervals, expressed or implied, being taken into consideration. It can be said to represent the beliefs in various sub-parts of the feasible intervals. As a result, total lack of overlap is not required for successful discrimination between alternatives. Rather, an overlap by interval parts carrying little belief mass, i.e. representing a small part of the agent's belief, is allowed. Then, the non-overlapping parts can be thought of as being the core of the agent's appreciation of the decision situation, thus allowing discrimination.

D

There are essentially three ways of evaluating (i.e. making a decision in) a second-order agent decision problem. The first way (centroid analysis) is to use the centroid as the best single-point representative of the distributions. The centroid is additive and multiplicative. Thus, the centroid of the distribution of expected utility is the expected utility of the centroids of the projections. A centroid analysis gives a good overview of a decision situation. The second way (contraction analysis) is to use the **centroid** as a focal point (contraction point) towards which the intervals are decreased while studying the overlap in first-order expected utility intervals. The third way (distribution analysis) is more elaborated, involving the analysis of the resulting distributions of expected utility and calculating the fraction of belief overlapping between alternatives being evaluated.

FUTURE TRENDS

During recent years, the activities within the area of imprecise probabilities have increased substantially (IPP) and special conferences (ISIPTA) and journals (IJAR) are now dedicated to this theme. Second-order theories will in the future be fully developed into functional theories of decision making suitable for dynamic use by distributed software components. Algorithms for the efficient evaluation by agents using at least the first two ways of analyses above will be developed.

CONCLUSION

In this article, we discuss various approaches to probabilistic decision making in agents. We point out that theories incorporating second-order belief can provide more powerful discrimination to the agent (software agent or human being) when handling aggregations of interval representations, such as in decision trees or probabilistic networks, and that interval estimates (upper and lower bounds) in themselves are not complete. This applies to all kinds of decision trees and probabilistic networks since they all use multiplications for the evaluations. The key idea is to use the information available in efficient evaluation of decision structures. Using only interval estimates often does not provide enough discrimination power for the agent to generate a preference order among alternatives considered.

Second-order methods are not just nice theories, but should be taken into account to provide efficient decision methods for agents, in particular when handling aggregations of imprecise representations as is the case in decision trees or probabilistic networks.

REFERENCES

Choquet, G. (1953/54). Theory of Capacities, *Ann. Inst. Fourier* 5, 131–295.

Danielson, M. & Ekenberg, L. (1998). A Framework for Analysing Decisions under Risk, *European Journal of Operational Research 104(3)*, 474–484.

Danielson, M. & Ekenberg, L. (2007). Computing Upper and Lower Bounds in Interval Decision Trees, *European Journal of Operational Research 181*, 808–816.

Danielson, M., Ekenberg, L., & Larsson, A. (2007). Belief Distribution in Decision Trees, to appear in *International Journal of Approximate Reasoning*, DOI 10.1016/j.ijar.2006.09.012.

Dempster, A.P. (1967). Upper and Lower Probabilities Induced by a Multivalued Mapping, *Annals of Mathematical Statistics xxxviii*, 325–339.

Denneberg, D. (1994). *Non-Additive Measure and Integral*, Kluwer Academic Publishers.

Ekenberg, L. (2000a). Risk Constraints in Agent Based Decisions, A. Kent & J. G. Williams eds., *Encyclopaedia of Computer Science and Technology 23:48*, 263–280, Marcel Dekker.

Ekenberg, L. (2000b). The Logic of Conflicts between Decision Making Agents, *Journal of Logic and Computation 10(4)*, 583–602.

Ekenberg, L., Boman, M., & Linneroth-Bayer, J. (2001). General Risk Constraints, *Journal of Risk Research 4(1)*, 31–47.

Ekenberg, L., Danielson, M., & Thorbiörnson, J. (2006). Multiplicative Properties in Evaluation of Decision Trees, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14(3)*, 293–316.

Ekenberg, L. & Thorbiörnson, J. (2001). Second-Order Decision Analysis, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(1)*, 13–38.

D

Ekenberg, L., Thorbiörnson, J., & Baidya, T. (2005). Value Differences using Second-order Distributions, *International Journal of Approximate Reasoning 38(1)*, 81–97.

Fischhoff, B., Goitein, B., & Shapira, Z. (1983). Subjective Expected Utility: A Model of Decision Making, *Decision making under Uncertainty*, R.W. Scholz, ed., Elsevier Science Publishers B.V. North-Holland, 183–207.

Gärdenfors, P. & Sahlin, N.E. (1982). Unreliable Probabilities, Risk Taking, and Decision Making, *Synthese 53*, 361–386.

Gärdenfors, P. & Sahlin, N.E. (1983). Decision Making with Unreliable Probabilities, *British Journal of Mathematical and Statistical Psychology 36*, 240–251.

Good, I.J. (1962). Subjective Probability as the Measure of a Non-measurable Set, *Logic, Methodology, and the Philosophy of Science*, Suppes, Nagel, & Tarski, eds., Stanford University Press, 319–329.

Herstein, I.N. & Milnor, J. (1953). An Axiomatic Approach to Measurable Utility, *Econometrica 21*, 291–297.

Hodges, J.L. & Lehmann, E.L. (1952). The Use of Previous Experience in Reaching Statistical Decisions, *The Annals of Mathematical Statistics 23*, 396–407.

Huber, P.J. (1973). The Case of Choquet Capacities in Statistics, *Bulletin of the International Statistical Institute 45*, 181–188.

Huber, P.J. & Strassen, V. (1973). Minimax Tests and the Neyman-Pearsons Lemma for Capacities, *Annals of Statistics 1*, 251–263.

Hurwicz, L. (1951). Optimality Criteria for Decision Making under Ignorance, *Cowles Commission Discussion Paper 370*.

International Journal of Approximate Reasoning (IJAR), http://www.sciencedirect.com.

Imprecise Probability Project (IPP), http://ippserv.rug.ac.be/home/ipp.html.

ISIPTA Conferences, http://www.sipta.org.

Jaffray, J-Y. (1999). Rational Decision Making With Imprecise Probabilities, Proceedings of ISIPTA'99.

Jeffrey, R. (1965/83). *The Logic of Decision*, 2nd ed., University of Chicago Press. (First edition 1965)

Kyburg, H.E. (1961). *Probability and the Logic of Rational Belief,* Connecticut: Wesleyan University Press.

Levi, I. (1974). On Indeterminate Probabilities, *The Journal of Philosophy 71*, 391–418.

Levi, I. (1980). *The Enterprise of Knowledge*, MIT Press.

Luce, R.D. & Krantz, D. (1971). Conditional Expected Utility, *Econometrica 39*, 253–271.

Nau, R.F. (2002). The aggregation of imprecise probabilities, *Journal of Statistical Planning and Inference 105*, 265–282.

von Neumann, J. & Morgenstern, O. (1947). *Theory of Games and Economic Behaviour*, 2nd ed., Princeton University Press.

Nilsson, N. (1986). Probabilistic Logic, *Artificial Intelligence 28*, 71–87.

Raiffa, H. (1968). *Decision Analysis*, Addison Wesley.

Ramsey, F.P. (1926/78). Truth and Probability, *Foundations: Essays in Philosophy, Logics, Mathematics and Economics*, ed. Mellor, 58–100, Routledge and Kegan Paul. (Originally from 1926)

Russell, S.J. & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Prentice-Hall.

Savage, L. (1954/72). *The Foundations of Statistics*, 2nd ed., John Wiley and Sons. (First edition 1954)

Shafer, G., Gillet, P.R. & Scherl, R.B. (2003). Subjective Probability and Lower and Upper Prevision: A New Understanding, Proceedings of ISIPTA 03.

Shafer, G. (1976). *A Mathematical Theory of Evidence*, Princeton University Press.

Shapira, Z. (1995). *Risk Taking: A Managerial Perspective*, Russel Sage Foundation.

Smith, C.A.B. (1961). Consistency in Statistical Inference and Decision, *Journal of the Royal Statistic Society, Series B xxiii*, 1–25.

Suppes, P. (1956). The Role of Subjective Probability and Utility Maximization, Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability 1954-55 5, 113–134.

Utkin, L.V. & Augustin,T. (2003). Decision Making with Imprecise Second-Order Probabilities, Proceedings of ISIPTA'03.

Wald, A. (1950). *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall.

Walley, P. (1991). *Statistical Decision Functions*, John Wiley and Sons.

Walley, P. (1997). Statistical inferences based on a second-order possibility distribution, *International Journal of General Systems 9*, 337–383.

Weichselberger, K. & Pöhlman, S. (1990). *A Methodology for Uncertainty in Knowledge-Based Systems*, Springer-Verlag.

Wilson, N. (1999). A Logic of Extended Probability, Proceedings of ISIPTA'99.

KEY TERMS

Admissible Alternative: Given a decision tree and two alternatives A_i and A_j, A_i is at least as good as A_j iff $E(A_i) - E(A_j) > 0$, where $E(A_i)$ is the expected value of A_i, for all consistent variable assignments for the probabilities and values. A_i is better than A_j iff A_i is at least as good as A_j and $E(A_i) - E(A_j) > 0$ for some consistent variable assignments for the probabilities and values. A_i is admissible iff no other A_j is better.

Centroid: Given a belief distribution F over a cube B, the centroid F_c of F is

$$F_c = \int_B x F(x) dV_B(x),$$

where V_B is some k-dimensional Lebesque measure on B.

Decision Tree: A decision tree consists of a root node, representing a decision, a set of intermediate (event) nodes, representing some kind of uncertainty and consequence nodes, representing possible final outcomes. Usually, probability distributions are assigned in the form of weights in the probability nodes as measures of the uncertainties involved.

Expected Value: Given a decision tree with r alternatives A_i for $i = 1,\ldots,r$, the expression

$$E(A_i) =$$
$$\sum_{i_1=1}^{n_{i_0}} p_{i i_1} \sum_{i_2=1}^{n_{i_1}} p_{i i_1 i_2} \cdots \sum_{i_{m-1}=1}^{n_{i_{m-2}}} p_{i i_1 i_2 \cdots i_{m-2} i_{m-1}} \sum_{i_m=1}^{n_{i_{m-1}}} p_{i i_1 i_2 \cdots i_{m-2} i_{m-1} i_m} v_{i i_1 i_2 \cdots i_{m-2} i_{m-1} i_m 1}$$

where $p_{\ldots i_j \ldots}$, $j \in (1,\ldots,m)$, denote probability variables and $v_{\ldots i_{j} \ldots 1}$ denote value variables, is the *expected value* of alternative A_i.

Joint Belief Distribution: Let a unit cube be represented by $B = (b_1,\ldots,b_k)$. By a joint belief distribution over B, we mean a positive distribution F defined on the unit cube B such that

$$\int_B F(x) dV_B(x) = 1,$$

where V_B is some k-dimensional Lebesque measure on B.

Marginal Belief Distribution: Let a unit cube:

$B = (b_1,\ldots,b_k)$ and $F \in BD(B)$ be given. Furthermore, let $B_i^- = (b_1,\ldots,b_{i-1},b_{i+1},\ldots,b_k)$. Then

$$f_i(x_i) = \int_{B_i^-} F(x) dV_{B_i^-}(x)$$

is a marginal belief distribution over the axis b_i.

Projection: Let $B = (b_1,\ldots,b_k)$ and $A = (b_{i_1},\ldots,b_{i_s})$:

$i_j \in \{1,\ldots k\}$ be unit cubes. Furthermore, let $F \in BD(B)$, and let

$$f_A(x) = \int_{B-A} F(x) dV_{B-A}(x) \cdot$$

Then f_A is the *projection of F on A*. A projection of a belief distribution is also a belief distribution.

Decision Tree Applications for Data Modelling

Man Wai Lee
Brunel University, UK

Kyriacos Chrysostomou
Brunel University, UK

Sherry Y. Chen[1]
Brunel University, UK

Xiaohui Liu
Brunel University, UK

INTRODUCTION

Many organisations, nowadays, have developed their own databases, in which a large amount of valuable information, e.g., customers' personal profiles, is stored. Such information plays an important role in organisations' development processes as it can help them gain a better understanding of customers' needs. To effectively extract such information and identify hidden relationships, there is a need to employ intelligent techniques, for example, data mining.

Data mining is a process of knowledge discovery (Roiger & Geatz, 2003). There are a wide range of data mining techniques, one of which is decision trees. Decision trees, which can be used for the purposes of classifications and predictions, are a tool to support decision making (Lee et al., 2007). As a decision tree can accurately classify data and make effective predictions, it has already been employed for data analyses in many application domains. In this paper, we attempt to provide an overview of the applications that decision trees can support. In particular, we focus on business management, engineering, and health-care management.

The structure of the paper is as follows. Firstly, Section 2 provides the theoretical background of decision trees. Section 3 then moves to discuss the applications that decision trees can support, with an emphasis on business management, engineering, and health-care management. For each application, how decision trees can help identify hidden relationships is described. Subsequently, Section 4 provides a critical discussion

of limitations and identifies potential directions for future research. Finally, Section 5 presents the conclusions of the paper.

BACKGROUND

Decision trees are one of the most widely used classification and prediction tools. This is probably because the knowledge discovered by a decision tree is illustrated in a hierarchical structure, with which the discovered knowledge can easily be understood by individuals even though they are not experts in data mining (Chang et al., 2007). A decision tree model can be created in several ways using existing decision tree algorithms. In order to effectively adopt such algorithms, there is a need to have a solid understanding of the processes of creating a decision tree model and to identify suitability of the decision tree algorithms used. These issues are described in subsections below.

Processes of Model Development

A common way to create a decision tree model is to employ a top-down, recursive, and divide-and-conquer approach (Greene & Smith, 1993). Such a modelling approach enables the most significant attribute to be located at the top level as a root node and the least significant attributes to be located at the bottom level as leave nodes (Chien et al., 2007). Each path between the root node and the leave node can be interpreted as an 'if-then' rule, which can be used for making predications (Chien et al., 2007; Kumar & Ravi, 2007).

To create a decision tree model on the basis of the above-mentioned approach, the modelling processes can be divided into three stages, which are: (1) tree growing, (2) tree pruning, and (3) tree selection.

Tree Growing

The initial stage of creating a decision tree model is tree growing, which includes two steps: tree merging and tree splitting. At the beginning, the non-significant predictor categorises and the significant categories within a dataset are grouped together (tree merging). As the tree grows, impurities within the model will increase. Since the existence of impurities may result in reducing the accuracy of the model, there is a need to purify the tree. One possible way to do it is to remove the impurities into different leaves and ramifications (tree splitting) (Chang, 2007).

Tree Pruning

Tree pruning, which is the key elements of the second stage, is to remove irrelevant splitting nodes (Kirkos et al., 2007). The removal of irrelevant nodes can help reduce the chance of creating an over-fitting tree. Such a procedure is particularly useful because an over-fitting tree model may result in misclassifying data in real world applications (Breiman et al., 1984).

Tree Selection

The final stage of developing a decision tree model is tree selection. At this stage, the created decision tree model will be evaluated by either using cross-validation or a testing dataset (Breiman *et al.*, 1984). This stage is essential as it can reduce the chances of misclassifying data in real world applications, and consequently, minimise the cost of developing further applications.

Suitability of Decision Tree Algorithms

A review of existing literature shows that the most widely used decision tree algorithms include the Iterative Dichotomiser 3 (ID3) algorithm, the C4.5 algorithm, the Chi-squared Automatic Interactive Detector (CHAID) algorithm, and the Classification and Regression Tree (CART) algorithm. Amongst these algorithms, there are some differences, one of which is the capability of

modelling different types of data. As a dataset may be constructed by different types of data, e.g., categorical data, numerical data, or the combination of both, there is a need to use a suitable decision tree algorithm which can support the particular type of data used in the dataset. All of the above-mentioned algorithms can support the modelling of categorical data whilst only the C4.5 algorithm and the CART algorithm can be used for the modelling of numerical data (see Table 1). This difference can also be used as a guideline for the selection of a suitable decision tree algorithm. The other difference amongst these algorithms is the process of model development, especially at the stages of tree growing and tree pruning. In terms of the former, the ID3 and C4.5 algorithms split a tree model into as many ramifications as necessary whereas the CART algorithm can only support binary splits. Regarding the latter, the pruning mechanisms located within the C4.5 and CART algorithms support the removal of insignificant nodes and ramifications but the CHAID algorithm hinders the tree growing process before the training data is being overused (see Table 1).

DECISION TREE APPLICATIONS

Business Management

In the past decades, many organizations had created their own databases to enhance their customer services. Decision trees are a possible way to extract useful information from databases and they have already been employed in many applications in the domain of business and management. In particular, decision tree modelling is widely used in customer relationship management and fraud detection, which are presented in subsections below.

Customer Relationship Management

A frequently used approach to manage customers' relationships is to investigate how individuals access online services. Such an investigation is mainly performed by collecting and analyzing individuals' usage data and then providing recommendations based on the extracted information. Lee et al. (2007) apply decision trees to investigate the relationships between the customers' needs and preferences and the success of online shopping. In their study, the frequency of us-

ing online shopping is used as a label to classify users into two categories: (a) users who rarely used online shopping and (b) users who frequently used online shopping. In terms of the former, the model suggests that the time customers need to spend in a transaction and how urgent customers need to purchase a product are the most important factors which need to be considered. With respect to the latter, the created model indicates that price and the degree of human resources involved (e.g. the requirements of contacts with the employees of the company in having services) are the most important factors. The created decision trees also suggest that the success of an online shopping highly depends on the frequency of customers' purchases and the price of the products. Findings discovered by decision trees are useful for understanding their customers' needs and preferences.

Fraudulent Statement Detection

Another widely used business application is the detection of Fraudulent Financial Statements (FFS). Such an application is particularly important because the existence of FFS may result in reducing the government's tax income (Spathis *et al.*, 2003). A traditional way to identify FFS is to employ statistical methods. However, it is difficult to discover all hidden information due to the necessity of making a huge number of assumptions and predefining the relationships among the large number of variables in a financial statement.

Previous research has proved that creating a decision tree is a possible way to address this issue as it can consider all variables during the model development process. Kirkos *et al.* (2007) have created a decision tree model to identify and detect FFS. In their study, 76 Greek manufacturing firms have been selected and their published financial statements, including balance sheets and income statements, have been collected for modelling purposes. The created tree model shows that all non-fraud cases and 92% of the fraud cases have been correctly classified. Such a finding indicates that decision trees can make a significant contribution for the detection of FFS due to a highly accurate rate.

Engineering

The other important application domain that decision trees can support is engineering. In particular, decision trees are widely used in energy consumption and fault diagnosis, which are described in subsections below.

Energy Consumption

Energy consumption concerns how much electricity has been used by individuals. The investigation of energy consumption becomes an important issue as it helps utility companies identify the amount of energy needed. Although many existing methods can be used for the investigation of energy consumption, decision trees appear to be preferred. This is due to the fact that

Table 1. Characteristics of different decision tree algorithms

Decision tree algorithms	Data types	Numerical data splitting method	Possible tool
CHAID (Kass, 1980)	Categorical	N/A	SPSS Answer Tree (SPSS Inc, 2007)
ID3 (Quinlan, 1986)	Categorical	No restrictions	WEKA (Ian and Eibe, 2005)
C4.5 (Quinlan, 1993)	Categorical, numerical	No restrictions	WEKA (Ian and Eibe, 2005)
CART (Breiman *et al.*, 1984)	Categorical, numerical	Binary splits	CART 5.0 (Salford Systems, 2004)

a hierarchical structure provided by decision trees is useful to present the deep level of information and insight. For instance, Tso and Yau (2007) create a decision tree model to identify the relationships between a household and its electricity consumptions in Hong Kong. Findings from their tree model illustrate that the number of household members are the most determinant factor of energy consumption in summer, and both the number of air-conditioner and the size of a flat are the second most important factors. In addition to such findings, their tree model identifies that a household with four or more members with a flat size larger than 817ft^2 is the highest electricity consumption group. On the other hand, households which have less than four family members and without air-conditioners are the smallest electricity consumption group. Such findings from decision trees not only provide a deeper insight of the electricity consumptions within an area but also give guidelines to electricity companies about the right time they need to generate more electricity.

Fault Diagnosis

Another widely used application in the engineering domain is the detection of faults, especially in the identification of a faulty bearing in rotary machineries. This is probably because a bearing is one of the most important components that directly influences the operation of a rotary machine. To detect the existence of a faulty bearing, engineers tend to measure the vibration and acoustic emission (AE) signals emanated from the rotary machine. However, the measurement involves a number of variables, some of which may be less relevant to the investigation. Decision trees are a possible tool to remove such irrelevant variables as they can be used for the purposes of feature selection. Sugumaran and Ramachandran (2007) create a decision tree model to identify the features that may significantly affect the investigation of a faulty bearing. Through feature selection, three attributes were chosen to discriminate the faulty conditions of a bearing, i.e., the minimum value of the vibration signal, the standard deviation of the vibration signal, and kurtosis. The chosen attributes, subsequently, were used for creating another decision tree model. Evaluations from this model show that more than 95% of the testing dataset has been correctly classified. Such a highly accurate rate suggests that the removal of insignificant attributes within a dataset is another contribution of decision trees.

Healthcare Management

As decision tree modelling can be used for making predictions, there are an increasing number of studies that investigate to use decision trees in health-care management. For instance, Chang (2007) has developed a decision tree model on the basis of 516 pieces of data to explore the hidden knowledge located within the medical history of developmentally-delayed children. The created model identifies that the majority of illnesses will result in delays in cognitive development, language development, and motor development, of which accuracies are 77.3%, 97.8%, and 88.6% respectively. Such findings can result in assisting healthcare professional to have an early intervention on developmentally-delayed children so as to help them catch up their normal peers in their development and growth. Another example of health-care management can be found in Delen *et al.*(2005). In their study, a decision tree is created to predict the survivability of breast cancer patients. The classification accuracy is 93.6% in their decision tree. This classification rate indicates that the created tree is highly accurate for predicting the survivability of breast cancer patients. These studies suggest that decision tree is a useful tool to discover and explore hidden information in health-care management.

FUTURE TRENDS

The applications domains mentioned above demonstrate that decision tree is a very useful tool for data analyses. However, there are still many limitations which we need to be aware of and addressed in future works.

Reliability of Findings

Although decision tree is a powerful tool for data analyses, it seems that some data are misclassified in the decision tree models. A possible way to address this issue is to exploit the extracted knowledge by human-computer collaboration. In other words, experts from different domains use their domain knowledge to filter findings from the created model. By doing so, the irrelevant findings can manually be removed. However, the drawback of employing such a method is the necessity of large investment as it involves the cost and time of experts from different domains.

Suitability of Algorithms

As described in Section 2.2, the development of a decision tree model involves the selection of an appropriate decision tree algorithm. In addition to taking into account the type of data being modelled, there is a need to consider the effectiveness of the algorithms. Another possible direction for future research is to compare the effectiveness of various algorithms and identify the strengths and weaknesses of each algorithm for different types of applications. In addition, it would be interesting for future research to conduct comparisons between decision tree algorithms and other types of classification algorithms. By doing so, guidelines for the selection of suitable decision tree algorithms for different types of applications can be generated.

CONCLUSION

The main objective of this paper is to help readers get an overall picture of decision trees by introducing its applications in different domains. To achieve this objective, this paper has provided an overview of the applications of decision tree modelling in business management, engineering, and health-care management domains. In each application domain, the benefits of creating a decision tree model for the purposes of analyzing data and making predictions have been identified. Such benefits include: (1) the capability to accurately discover hidden relationships between variables, (2) the presentation of knowledge in a deep level of understanding and insight on the basis of its hierarchical structure, and (3) the capability of removing insignificant attributes within a dataset.

Three application domains have been studied in this paper, but it ought to be noted that decision trees can also be applied in other application domains, e.g. bioinformatics and psychology. These application domains should also be examined. The findings of such studies can subsequently be integrated into those of this study so that a complete framework for implementing decision trees models can be created. Such a framework would be useful to enhance the depth and breadth of the knowledge of decision tree models.

REFERENCES

Bevilacqua, M., Braglia, M., & Montanari, R. (2003). The Classification and Regression Tree Approach to Pump Failure Rate Analysis. *Reliability Engineering and System Safety , 79* (1), 59-67.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees.* Pacific Grove: Wadsworth.

Chang, C. L. (2007). A Study of Applying Data Mining to Early Intervention for Developmentally-dalayed Children. *Expert Systems with Applications , 33* (2), 407-412.

Chang, Y. C., Lai, P. C., & Lee, M. T. (2007). An Integrated Approach for Operational Knowledge Acquisition of Refuse Incinerators. *Expert Systems with Applications , 33* (2), 413-419.

Chien, C. F., Wang, W. C., & Cheng, J. C. (2007). Data Mining for Yield Enhancement in Semiconductor Manufacturing and An Empirical Study. *Expert Systems with Applications , 33* (1), 192-198.

Delen, D., Walker, G., & Kadam, A. (2005). Predicting Breast Cancer Survivability: A Comparison of Three Data Mining Methods. *Artificial Intelligence In Medicine , 34* (2), 113-127.

Greene, D. P., & Smith, S. F. (1993). Competition-based Induction of Decision Models from Example. *Machine Learning , 13* (2-3), 229-257.

Ian, H. W., & Eibe F. (2005). *Data Mining: Practical Machine Learning Tools And Techniques.* 2nd Edition, San Francisco: Morgan Kaufmann.

Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics , 29*, 119-127.

Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data Mining Techniques for the Detection of Fraudulent Financial Statements. *Expert Systems with Applications, 32* (4), 995-1003.

Kumar, P. R., & Ravi, V. (2007). Bankruptcy Prediction in Banks and Firms via Statistical and Intelligent Techniques - a Review. *European Journal of Operational Research , 180* (1), 1-28.

D

Lee, S., Lee, S., & Park, Y. (2007). A Prediction Model for Success of Services in E-commerce Using Decision Tree: E-customer's Attitude Towards Online Service. *Expert Systems with Applications , 33* (3), 572-581.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning , 1* (1), 81-106.

Roiger, R. J., & Geatz, M. W. (2003). *Data Mining: A Tutorial-Baseed Primer.* Montreal: Addison-Wesley.

Salford Systems (2004) CART 5.0 [WWW page]. URL http://www.salfordsystems.com/cart.php

Spathis, C., Doumpos, M., & Zopounidis, C. (2003). Using Client Performance Measures to Identify Pre-engagement Factors Associated with Qualified Audit Reports in Greece. *The International Journal of Accounting , 38* (3), 267-284.

SPSS Inc. (2007). Target the Right People More Effectively with AnswerTree [WWW page]. URL http://www.spss.com/answertree

Sugumaran, V., & Ramachandran, K. (2007). Automatic Rule Learning Using Decision Tree for Fuzzy Classifier in Fault Diagnosis of Roller Bearing. *Mechanical Systems and Signal Processing , 21* (5), 2237-2247.

Sun, W., Chen, J., & Li, J. (2007). Decision Tree and PCA-based Fault Diagnosis of Rotating Machinery. *Mechanical Systems and Signal Processing , 21* (3), 1300-1317.

Tso, G. K., & Yau, K. K. (2007). Predicting Electricity Energy Consumption: A Comparison of Regression Analysis, Decision Tree and Neural Networks. *Energy*.

KEY TERMS

Attributes: Pre-defined variables in a dataset.

Classification: An allocation of items or objects to classes or categories according to their features.

Customer Relationship Management: A dynamic process to manage the relationships between a company and her customers, including collecting, storing and analysing customers' information.

Data Mining: Also known as knowledge discovery in database (KDD), which is a process of knowledge discovery by analysing data and extracting information from a dataset using machine learning techniques.

Decision Tree: A predictive model which can be visualized in a hierarchical structure using leaves and ramifications.

Decision Tree Modelling: The process of creating a decision tree model.

Fault Diagnosis: An action of identifying a malfunctioning system based on observing its behaviour.

Fraud Detection Management: The detection of frauds, especially in those existing in financial statements or business transactions so as to reduce the risk of loss.

Healthcare Management: The act of preventing, treating and managing illness, including the preservation of mental and physical problems through the services provided by health professionals.

Prediction: A statement or a claim that a particular event will happen in the future.

The Dempster–Shafer Theory

D

Malcolm J. Beynon
Cardiff University, UK

INTRODUCTION

The initial work introducing Dempster-Shafer (D-S) theory is found in Dempster (1967) and Shafer (1976). Since its introduction the very name causes confusion, a more general term often used is belief functions (both used intermittently here). Nguyen (1978) points out, soon after its introduction, that the rudiments of D-S theory can be considered through distributions of random sets. More furtive comparison has been with the traditional Bayesian theory, where D-S theory has been considered a generalisation of it (Schubert, 1994). Cobb and Shenoy (2003) direct its attention to the comparison of D-S theory and the Bayesian formulisation. Their conclusions are that they have the same expressive power, but that one technique cannot simply take the role of the other.

The association with artificial intelligence (AI) is clearly outlined in Smets (1990), who at the time, acknowledged the AI community has started to show interest for what they call the Dempster-Shafer model. It is of interest that even then, they highlight that there is confusion on what type of version of D-S theory is considered. D-S theory was employed in an event driven integration reasoning scheme in Xia *et al*. (1997), associated with automated route planning, which they view as a very important branch in applications of AI. Liu (1999) investigated Gaussian belief functions and specifically considered their proposed computation scheme and its potential usage in AI and statistics. Huang and Lees (2005) apply a D-S theory model in natural-resource classification, comparing with it with two other AI models.

Wadsworth and Hall (2007) considered D-S theory in a combination with other techniques to investigate site-specific critical loads for conservation agencies. Pertinently, they outline its positioning with respect to AI (p. 400);

The approach was developed in the AI (artificial intelligence) community in an attempt to develop systems that could reason in a more human manner and par-ticularly the ability of human experts to "diagnose" situations with limited information.

This statement is pertinent here, since emphasis within the examples later given is more towards the general human decision making problem and the handling of ignorance in AI. Dempster and Kong (1988) investigated how D-S theory fits in with being an artificial analogy for human reasoning under uncertainty.

An example problem is considered, the murder of Mr. White, where witness evidence is used to classify the belief in the identification of an assassin from considered suspects. The numerical analyses presented exposit a role played by D-S theory, including the different ways it can act on incomplete knowledge.

BACKGROUND

The background section to this article covers the basic formulisations of D-S theory, as well as certain developments. Formally, D-S theory is based on a finite set of p elements $\Theta = \{s_1, s_2, ..., s_p\}$, called a frame of discernment. A *mass value* is a function m: $2^\Theta \rightarrow [0, 1]$ such that $m(\varnothing) = 0$ ($\varnothing$ - the empty set) and:

$$\sum_{s \in 2^\Theta} m(s) = 1$$

(2^Θ - the power set of Θ). Any proper subset s of the frame of discernment Θ, for which $m(s)$ is non-zero, is called a focal element and represents the exact belief in the proposition depicted by s. The notion of a proposition here being the collection of the hypotheses represented by the elements in a focal element.

In the original formulisation of D-S theory, from a single piece of evidence all assigned mass values sum to unity and there is no belief in the empty set. In the case of the Transferable Belief Model (TBM), a fundamental development on the original D-S theory (see Smets and Kennes, 1994), a non-zero mass value

can be assigned to the empty set allowing $m(\varnothing) \geq 0$. The set of mass values associated with a single piece of evidence is called a *body of evidence* (BOE), often denoted $m(\cdot)$. The mass value $m(\Theta)$ assigned to the frame of discernment Θ is considered the amount of ignorance within the BOE, since it represents the level of exact belief that cannot be discerned to any proper subsets of Θ.

D-S theory also provides a method to combine the BOE from different pieces of evidence, using Dempster's rule of combination. This rule assumes these pieces of evidence are independent, then the function $(m_1 \oplus m_2): 2^\Theta \to [0, 1]$, defined by:

$$(m_1 \oplus m_2)(x) =$$

$$\begin{cases} 0 & x = \varnothing \\ \dfrac{\sum\limits_{s_1 \cap s_2 = x} m_1(s_1) m_2(s_2)}{1 - \sum\limits_{s_1 \cap s_2 = \varnothing} m_1(s_1) m_2(s_2)} & x \neq \varnothing \end{cases}$$

$$(1)$$

is a mass value, where s_1 and s_2 are focal elements from the BOEs, $m_1(\cdot)$ and $m_2(\cdot)$, respectively. The denominator part of the combination expression includes:

$$\sum_{s_1 \cap s_2 = \varnothing} m_1(s_1) m_2(s_2)$$

that measures the level of conflict in the combination process (Murphy, 2000). It is the existence of the denominator part in this combination rule that separates D-S theory (includes it) from TBM (excludes it). Benouhiba and Nigro (2006) view this difference as whether considering the conflict mass:

$$\left(\sum_{s_1 \cap s_2 = \varnothing} m_1(s_1) m_2(s_2) \right)$$

as a further form of ignorance mass is an acceptable point of view.

D-S theory, along with TBM, also differs to the Bayesian approach in that it does not necessarily produce final results. Moreover, partial answers are present in the final BOE produced (through the combination of evidence), including focal elements with more than one element, unlike the Bayesian approach where probabilities on only individual elements would be accrued. This restriction of the Bayesian approach to consider singleton elements is clearly understood through the 'Principle of insufficient Reason', see Beynon *et al.* (2000) and Beynon (2002, 2005).

To enable final results to be created with D-S theory, a number of concomitant functions exist with D-S theory, including;

i)   The Belief function,

$$\text{Bel}(s_i) = \sum_{s_j \subseteq s_i} m(s_j)$$

for all $s_i \subseteq \Theta$, representing the confidence that a proposition y lies in s_i or any subset of s_i,

ii)   The Plausibility function,

$$\text{Pls}(s_i) = \sum_{s_j \cap s_i \neq \varnothing} m(s_j)$$

for all $s_i \subseteq \Theta$, represents the extent to which we fail to disbelieve s_i,

iii)   The Pignistic function (see Smets and Kennes, 1994),

$$\text{BetP}(s_i) = \sum_{s_j \subseteq \Theta, s_j \neq \varnothing} m(s_j) \frac{|s_i \cap s_j|}{|s_j|},$$

for all $s_i \subseteq \Theta$, represents the extent to which we fail to disbelieve s_i.

From the definitions given above, the Belief function is cautious of the ignorance incumbent in the evidence, where as the Plausibility function is more inclusive of its presence. The Pignistic function acts more like a probability function, partitioning levels of exact belief (mass) amongst the elements of the focal element it is associated with.

A non-specificity measure $N(m(\cdot))$ within D-S theory was introduced by Dubois and Prade (1985), the formula is defined as,

$$N(m(\cdot)) = \sum_{s_j \subseteq 2^\Theta} m(s_j) \log_2 |s_j|,$$

where $|s_j|$ is the number of elements in the focal element s_j. Hence, $N(m(\cdot))$ is considered the weighted average of the focal elements, with $m(\cdot)$ the degree of evidence focusing on s_j, while $\log_2 |s_j|$ indicates the lack of specificity of this evidential claim. The general range of this measure is $[0, \log_2|\Theta|]$ (given in Klir and Wierman, 1998), where $|\Theta|$ is the number of elements in the frame of discernment Θ.

Main Thrust

The main thrust of this article is an exposition of the utilisation of D-S theory. The small example problem considered here relates to the assassination of Mr White, many derivatives of this example exist. An adaptation of a version of this problem given in Smets (1990) is discussed, more numerical based here, which allows interpretation with D-S theory and its development TBM to be made.

There are three individuals who are suspects for the murder of Mr. White, namely, Henry, Tom and Sarah, within D-S theory they make up the frame of discernment, $\Theta = \{Henry, Tom, Sarah\}$. There are two witnesses who have information regarding the murder of Mr. White;

Witness 1, is 80% sure that the murderer was a man, it follows, the concomitant body of evidence (BOE), defined $m_1(\cdot)$, includes $m_1(\{Henry, Tom\}) = 0.8$. Since we know nothing about the remaining mass value it is considered ignorance, and allocated to Θ, hence $m_1(\{Henry, Tom, Sarah\}) = 0.2 (= m_1(\Theta))$.

Witness 2, is 60% confident that Henry was leaving on a jet plane when the murder occurred, so a BOE defined $m_2(\cdot)$ includes, $m_2(\{Tom, Sarah\}) = 0.6$ and $m_2(\{Henry, Tom, Sarah\}) = 0.4$.

The aggregation of these two sources of information (evidence from the two witnesses), using Dempster's combination rule (1), is based on the intersection and

Table 1. Intermediate combination of BOEs, $m_1(\cdot)$ and $m_2(\cdot)$

$m_1(\cdot) \setminus m_2(\cdot)$	{Tom, Sarah}, 0.6	Θ, 0.4
{Henry, Tom}, 0.8	{Tom}, 0.48	{Henry, Tom}, 0.32
Θ, 0.2	{Tom, Sarah}, 0.12	Θ, 0.08

multiplication of the focal elements and mass values from the BOEs, $m_1(\cdot)$ and $m_2(\cdot)$, see Table 1.

In Table 1, the intersection and multiplication of the focal elements and mass values from the BOEs, $m_1(\cdot)$ and $m_2(\cdot)$ are presented. The new focal elements found are all non-empty, it follows, the level of conflict

$$\sum_{s_1 \cap s_2 = \varnothing} m_1(s_1) m_2(s_2) = 0,$$

then the resultant BOE, defined $m_3(\cdot)$, can be taken directly from the results in Table 1;

$m_3(\{Tom\}) = 0.48$, $m_3(\{Henry, Tom\}) = 0.32$, $m_3(\{Tom, Sarah\}) = 0.12$ and $m_3(\{Henry, Tom, Sarah\}) = 0.08$.

Amongst this combination of evidence $(m_3(\cdot))$, the mass value assigned to ignorance $(m_3(\{Henry, Tom, Sarah\}) = 0.08)$ is less than that present in the original constituent BOEs, as expected when combining evidence using D-S theory. To further exposit the effect of the combination of evidence, the respective non-specificity values associated with BOEs shown here are calculated. For the two witnesses, with their BOEs, $m_1(\cdot)$ and $m_2(\cdot)$;

$$N(m_1(\cdot)) = \sum_{s_j \subseteq 2^\Theta} m_1(s_j) \log_2 |s_j|$$

$= m_1(\{Henry, Tom\}) \log_2|\{Henry, Tom\}|$
$\quad + m_1(\{Henry, Tom, Sarah\}) \log_2|\{Henry, Tom, Sarah\}|,$
$= 0.8 \times \log_2 2 + 0.2 \times \log_2 3 = 1.117,$

and $N(m_2(\cdot)) = 1.234$. The non-specificity associated with the combined is similarly calculated, found to be $N(m_3(\cdot)) = 0.567$. The values further demonstrate the

$$+\ m_3(\Theta)\frac{|\{Henry\}\cap\Theta|}{|\Theta|},$$

$$= 0.16 + 0.027 = 0.187,$$
similarly, BetP({Tom}) = 0.727 and BetP({Sarah})
$$= 0.087.$$

In this small example, all three measures identify the suspect Tom as having the most evidence purporting to them being the murderer of Mr. White.

FUTURE TRENDS

Dempster-Shafer (D-S) theory is a methodology that offers an alternative, possibly developed generality, to the assignment of frequency-based probability to events, in its case levels of subjective belief. However, the issues surrounding its position with respect to other methodologies such as the more well known Bayesian approach could be viewed as stifling it utilisation. The important point to remember when considering D-S theory is that it is a general methodology that requires subsequent pertinent utilisation when deriving nascent techniques.

Future work needs to aid in finding the position of D-S theory relative to the other methodologies. That is, unlike methodologies like fuzzy set theory, D-S theory is not able to be employed straight on top of existing techniques, to create a D-S type derivative of the technique. Such derivatives, for example, could operate on incomplete data, including when there are missing values, their reason for missing possibly due to ignorance etc.

CONCLUSION

Dempster-Shafer (D-S) theory, and its general developments, continues to form the underlying structure to an increasing number of specific techniques that attempt to solve certain problems within the context of uncertain reasoning. As mentioned in the future trends section, the difficulty with D-S theory is that it needs to be considered at the start of work at creating a new technique for analysis. It follows, articles like this which show the rudimentary workings of D-S theory allow researchers the opportunity to see its operation, and so may contribute to its further utilisation.

REFERENCES

Benouhiba, T., & Nigro, J.-M. (2006). An evidential cooperative multi-agent system, *Expert Systems with Applications*. *30*, 255-264.

Beynon, M.J. (2002). DS/AHP method: A mathematical analysis, including an understanding of uncertainty. *European Journal of Operational Research*, *140*(1), 149-165.

Beynon, M.J. (2005). Understanding Local Ignorance and Non-specificity in the DS/AHP Method of Multi-criteria Decision Making. *European Journal of Operational Research, 163*, 403-417.

Beynon, M., Curry, B., & Morgan, P. (2000). The Dempster-Shafer Theory of Evidence: An Alternative Approach to Multicriteria Decision Modelling. *OMEGA - International Journal of Management Science*, *28*(1), 37-50.

Cobb, B.R., & Shenoy, P.P. (2003). A Comparison of Bayesian and Belief Function Reasoning. *Information Systems Frontiers*, *5*(4), 345-358.

Dempster, A.P. (1967). Upper and lower probabilities induced by a multi-valued mapping. *Annals of Mathematical Statistics*, *38*, 325-339.

Dempster, A.P., & Kong, A. (1988). Uncertain evidence and artificial analysis. *Journal of Statistical Planning and Inference*, *1*, 355-368.

Dubois, D., & Prade, H. (1985). A note on measures of specificity for fuzzy sets. *International Journal of General Systems*, *10*, 279-283.

Huang, Z., & Lees, B. (2005). Representing and reducing error in natural-resource classification using model combination. *International Journal of Geographical Information Science*, *19*(5), 603-621.

Klir, G.J., & Wierman, M.J. (1998). *Uncertainty-Based Information: Elements of Generalized Information Theory*. Physica-Verlag, Heidelberg.

Liu, L. (1999). Local computation of Gaussian belief functions. *International Journal of Approximate Reasoning*, *22*, 217-248.

Murphy, C.K. (2000). Combining belief functions when evidence conflicts. *Decision Support Systems*, *29*, 1-9.

Nguyen, H.T. (1978). On random sets and belief functions. *Journal of Mathematical Analysis Applications*, *65*, 531-542.

Schubert, J. (1994). *Cluster-based specification techniques in Dempster-Shafer theory for an evidential intelligence analysis of multiple target tracks*. Department of Numerical Analysis and Computer Science Royal Institute of technology, S-100 44 Stockholm, Sweden.

Shafer, G.A. (1976). *Mathematical Theory of Evidence*. Princeton University Press, Princeton.

Smets, P. (1990). The Combination of Evidence in the Transferable belief Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(5), 447-458.

Smets, P., & Kennes, R. (1994). The transferable belief model. *Artificial Intelligence*, *66*(2), 191-243.

Xia, Y., Iyengar, S.S., & Brener, N.E. (1997). An event driven integration reasoning scheme for handling dynamic threats in an unstructured environment. *Artificial Intelligence*, *95*, 169-186.

Wadsworth, R.A., & Hall, J.R. (2007). Setting Site Specific Critical Loads: An Approach using Endorsement Theory and Dempster–Shafer. *Water Air Soil Pollution: Focus*, *7*, 399-405.

KEY TERMS

Belief: In Dempster-Shafer theory, the level of representing the confidence that a proposition lies in a focal element or any subset of it.

Body of Evidence: In Dempster-Shafer theory, a series of focal elements and associated mass values.

Focal Element: In Dempster-Shafer theory, a set of hypotheses with positive mass value in a body of evidence.

Frame of Discernment: In Dempster-Shafer theory, the set of all hypotheses considered.

Dempster-Shafer Theory: General methodology, also known as the theory of belief functions, its rudiments are closely associated with uncertain reasoning.

Ignorance: In Dempster-Shafer theory, the level of mass value not discernible among the hypotheses.

Mass Value: In Dempster-Shafer theory, the level of exact belief in a focal element.

Non-Specificity: In Dempster-Shafer theory, the weighted average of the focal elements' mass values in a body of evidence, viewed as a species of a higher uncertainty type, encapsulated by the term ambiguity.

Plausibility: In Dempster-Shafer theory, the extent to which we fail to disbelieve a proposition lies in a focal element.

Dependency Parsing: Recent Advances

D

Ruket Çakıcı
ICCS School of Informatics,University of Edinburgh, UK

INTRODUCTION

Annotated data have recently become more important, and thus more abundant, in computational linguistics . They are used as training material for machine learning systems for a wide variety of applications from Parsing to Machine Translation (Quirk et al., 2005). Dependency representation is preferred for many languages because linguistic and semantic information is easier to retrieve from the more direct dependency representation. Dependencies are relations that are defined on words or smaller units where the sentences are divided into its elements called heads and their arguments, e.g. verbs and objects. Dependency parsing aims to predict these dependency relations between lexical units to retrieve information, mostly in the form of semantic interpretation or syntactic structure.

Parsing is usually considered as the first step of Natural Language Processing (NLP). To train statistical parsers, a sample of data annotated with necessary information is required. There are different views on how informative or functional representation of natural language sentences should be. There are different constraints on the design process such as: 1) how intuitive (natural) it is, 2) how easy to extract information from it is, and 3) how appropriately and unambiguously it represents the phenomena that occur in natural languages.

In this article, a review of statistical dependency parsing for different languages will be made and current challenges of designing dependency treebanks and dependency parsing will be discussed.

DEPENDENCY GRAMMAR

The concept of dependency grammar is usually attributed to Tesnière (1959) and Hays (1964). The dependency theory has since developed, especially with the works of Gross (1964), Gaiffman (1965), Robinson (1970), Mel'čuk (1988), Starosta (1988), Hudson (1984, 1990), Sgall et al. (1986), Barbero et al.

(1998), Duchier (2001), Menzel and Schröder (1998), Kruijff (2001).

Dependencies are defined as links between lexical entities (words or morphemes) that connect heads and their dependants. Dependencies may have labels, such as *subject, object*, and *determiner* or they can be unlabelled. A dependency tree is often defined as a directed, acyclic graph of links that are defined between words in a sentence. Dependencies are usually represented as trees where the root of the tree is a distinct node. Sometimes dependency links cross. Dependency graphs of this type are non-projective. Projectivity means that in surface structure a head and its dependants can only be separated by other dependants of the same head (and dependants of these dependants). Non-projective dependency trees cannot be translated to phrase structure trees unless treated specially. We can see in Table 1 that the notion of non-projectivity is very common across languages although distribution of it is usually rare in any given language. The fact that it is rare does not make it less important because it is this kind of phenomena that makes natural languages more interesting and that makes all the difference in the generative capacity of a grammar that is suggested to explain natural languages.

An example dependency tree is in Figure 1. The corresponding phrase structure tree is shown in Figure 2. The ROOT of this tree is "*hit*".

Given the basic concept of dependency, different theories of dependency grammar exist. Among many well known are: Functional Generative Description (Sgall et al., 1969, 1986), (Petkevič, 1987, 1995), Dependency Unification Grammar (DUG) Hellwig (1986, 2003), Meaning Text Theory (Gladkij and Mel'čuk, 1975), (Mel'čuk, 1988) and Lexicase (Starosta, 1988), Topological Dependency Grammar (Gerdes and Kahane, 2001). Kruijff (2001) also suggests a type of logic for dependency grammar, "Dependency Grammar Logic" which aims transparent semantic interpretation during parsing.

There are many open issues regarding the representation of dependency structure. Hays (1964)

Figure1. Dependency Tree for the sentence "The red car hit the big motorcycle"
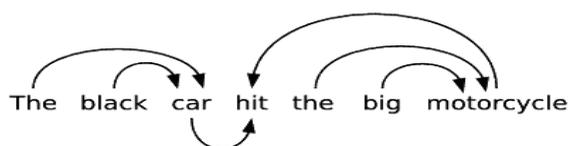
Figure 2. Phrase Structure Tree for the sentence in Figure 1
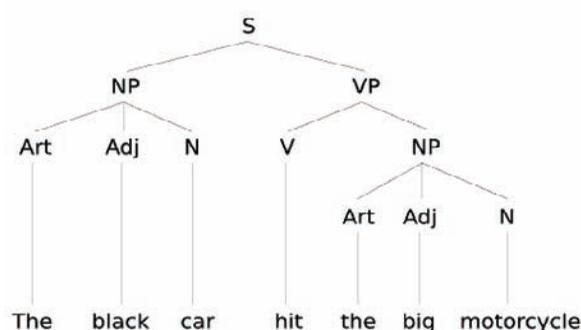
and Gaifman (1965) take dependency grammars as special cases of phrase structure grammars whereas Barbero et al. (1998), Menzel and Schröder (1998), Eisner (2000), Samuelsson (2000), Duchier (2001), Gerdes and Kahane (2001), Kruijff (2001) think they are completely different.

Generative capacity of dependency grammars has long been discussed (Gross, 1964), (Hays, 1964), (Gaifman, 1965), (Robinson, 1970). Dependency grammars were proved to be context-free (Gaiffman, 1965). When natural languages were proved to be not context-free, but in a class called "Mildly Context-Sensitive" (Joshi, 1985) they were abandoned until 90s, when Vijayashanker and Weir (1994) showed that Head Grammars -an extension of CFGs- (Pollard, 1984) are mildly context-sensitive like Tree Adjoining Grammar (TAG), (Joshi et al., 1975) and Combinatory Categorial Grammar (CCG), (Steedman,2000). Recently, Kuhlmann and Möhl (2007) defined "regular dependency languages" and showed that applying different combinations of gap-degree and well-nestedness restrictions on non-projectivity in these languages gave a class of mildly context-sensitive grammars.

DEPENDENCY TREEBANKS

Why Dependency Trees?

Many new corpora have been designed and created in the past few years. Dependency representation is preferred when these corpora are designed. This can be argued by the following properties of dependency trees:

1. They are easier to annotate than some other representation types like phrase structure trees (PST). There are fewer tags and labels (only as many as words in a sentence) and no internal nodes to name the phrases as in PSTs.
2. Some information such as predicate-argument structure can be extracted trivially from them which is not the case for PSTs.
3. Another interesting result is that some dependency parsers run much faster than PST parsers. Computational complexity of a standard PST parser is $O(n^5)$ whereas a non-projective DT parser runs in $O(n^2)$.

Dependency Treebanks

Table 1 compares dependency corpora of 19 languages[1]. This information is gathered from CoNLL-X and CoNNL 2007 shared tasks on dependency parsing. The reader is referred to Buchholz and Marsi (2006) and Nivre et al. (2007) for more information on dependency treebanks included in the tasks. Although, the underlying theory is the same in all of these treebanks there are major differences in the outcome that originate from the questions like 1) how much information is needed to put in the dependency trees, 2) how strongly interlaced the different modules such as morphology syntax are in a language. For instance, Czech treebank (Böhmová et al., 2003) has 3 different levels of representation, namely,

morphological, grammatical and tecto-grammatical layers. Morphology-syntax interface in Turkish makes word-based dependencies inappropriate (Çakıcı, 2008). Therefore, dependencies are between morphological sub-groups called inflectional groups (IG) rather than words. These are two arguments among many on why it is very important to make a good feasibility study when designing a dependency treebank as different aspects of languages require different treatment.

DEPENDENCY PARSING

Statistical or data-driven parsing methods have gained more focus with the continuous introduction of new

Table 1. Treebank information; #T = number of tokens * 1000, #S = number of sentences * 1000, #T/#S = tokens per sentence, %NST = % of non-scoring tokens (only in CoNLL-X), %NPR = % of non-projective relations, %NPS = % of non-projective sentences, IR = has informative root labels

Language	#T		#S		#T/#S		%NST		%NPR		%NPS		IR	
Arabic	54	112	1.5	2.9	37.2	38.3	8.8	-	0.4	0.4	11.2	10.1	Y	-
Basque	-	51	-	3.2	-	38.3	-	-	-	2.9	-	26.2	-	-
Bulgarian	190		12.8	-	14.8	-	14.4	-	0.4	-	5.4	-	N	-
Catalan	-	431	-	15	-	28.8	-	-	-	0.1	-	2.9	-	-
Chinese	337	337	57	57	5.9	5.9	0.8	-	0.0	0.0	0.0	0.0	N	-
Czech		432	72.7	25.4	17.2	17.0	14.9	-	1.9	1.9	23.2	23.2	Y	-
Danish	94	-	5.2	-	18.2	-	13.9	-	1.0	-	15.6	-	N	-
Dutch	195	-	13.3	-	14.6	-	11.3	-	5.4	-	36.4	-	N	-
English	-	447	-	18.6	-	24.0	-	-	-	0.3	-	6.7	-	-
German	700	-	39.2	-	17.8	-	11.5	-	2.3	-	27.8	-	N	-
Greek	-	65	-	2.7	-	24.2	-	-	-	1.1	-	20.3	-	-
Hungarian	-	132	-	6.0	-	21.8	-	-	-	2.9	-	26.4	-	-
Italian	-	71	-	3.1	-	22.9	-	-	-	0.5	-	7.4	-	-
Japanese	151	-	17	-	8.9	-	11.6	-	1.1	-	5.3	-	N	-
Portuguese	207	-	9.1	-	22.8	-	14.2	-	1.3	-	18.9	-	Y	-
Slovene	29	-	1.5	-	18.7	-	17.3	-	1.9	-	22.2	-	Y	-
Spanish	89	-	3.3	-	27	-	12.6	-	0.1	-	1.7	-	N	-
Swedish	91	-	11	-	17.3	-	11.0	-	1.0	-	9.8	-	N	-
Turkish	58	65	5	5.6	11.5	11.6	33.1	-	1.5	5.5	11.6	33.3	N	-

linguistic data. Parsing was more focused on training and parsing with phrase structure trees and specifically English language because the Penn Treebank (Marcus et al., 1993) was the only available source for a long time. With the introduction of treebanks of different languages it is now possible to explore the bounds of multilingual parsing.

The early efforts of data-driven dependency parsing were focused on translating dependency structures to phrase structure trees for which the parsers already existed. But it was realised quickly that doing this was not as trivial as previously thought. It is much more trivial and more intuitive to represent some phenomena with dependency trees rather than phrase structure trees such as local and global scrambling, in other words free word-order. Thus the incompatible translations of dependency structures to phrase structure trees resulted in varying degrees of loss of information.

Collins et al. (1999) reports results on Czech. He translates the dependency trees to phrase structure trees in the flattest way possible and names the internal nodes after part of speech tag of the head word of that node. He uses Model 2 in Collins (1999) and then evaluates the attachment score on the dependencies extracted from the resulting phrase structure trees of his parser. However, crossing dependencies cannot be translated into phrase structure trees (Çakıcı and Baldridge, 2006) unless surface order of the words is changed. But Collins et al. (1999) does not mention crossing dependencies, therefore, we do not know how he handled non-projectivity.

One of the earliest statistical systems that aims parsing dependency structures directly without an internal representation of translation is Eisner (1996). He proposes 3 different generative models. He evaluates them on the dependencies derived from the Wall Street Journal part of the Penn Treebank. Eisner reports 90 percent for probabilistic parsing of English samples from WSJ. He reports 93 percent attachment score when gold standard tags were used, which means 93 percent of all the dependencies are correct regardless of the percentage of the dependencies in each sentence. Eisner's parser is a projective parser thus it cannot inherently predict crossing dependencies.

Discriminative dependency parsers such as Kudo and Matsumoto (2000, 2002), and Yamada and Matsumoto (2003) were also developed. They use support vector machines to predict the next action of a deterministic

parser. Nivre et al. (2004) does this by memory-based learning. They are all deterministic parsers.

McDonald et al. (2005b) tried something new and applied graph spanning algorithms to dependency parsing. They formalise dependency parsing as the problem of finding a maximum spanning tree in a directed graph. MIRA is used to determine the weights of dependency links as part of this computation. This algorithm has two major advantages: it runs in $O(n^2)$ time and it can handle non-projective dependencies directly. They show that this algorithm significantly improves performance on dependency parsing for Czech, especially on sentences which contain at least one crossed dependency. Variations of this parser has been used in CoNLL-X shared task and received the highest ranking among the participants averaged over the results of all of the 13 languages (Buchholz and Marsi, 2006). However, when no linguistic or global constraints are applied it may yield absurd dependency sequences such as assigning two subjects to a verb (Riedel et al., 2006). McDonald (2005a) uses MIRA learning algorithm with Eisner's parser and reports results for projective parsing.

FUTURE TRENDS

There is growing body of work on creating new treebanks for different languages. Requirements for the design of these treebanks are at least as diverse as these natural languages themselves. For instance, some languages have a much more strong morphological component or freer word order than others. Understanding and modelling these in the form of annotated linguistic data will guide the understanding of natural language, and technological advancement will hopefully make it easier to understand the inner workings of the language faculty of humans. There are challenges both for dependency parsing and for the dependency theory. For instance, modelling long-distance dependencies and, multiple head dependencies are still awaiting attention and there is much to do on morphemic dependency approach where heads of phrases can be morphemes rather than words in a sentence for morphologically complex languages. Although these constitute a fraction of all the phenomena in natural languages, they are the "tricky" part of the NLP systems that will never be perfect as long as these natural phenomena are ignored.

D

CONCLUSION

This article has reviewed dependency grammar theory together with recent advances in statistical dependency parsing for different languages. Some current challenges in building dependency treebanks and dependency parsing have also been discussed. Dependency theory and practical applications of dependency representations have advantages and disadvantages. The fact that dependency parsing is easy to adapt to new languages, and is well-adapted to representing free word-order, makes it the preferred representation for many new linguistic corpora. Dependency parsing is also developing in the direction of multi-lingual parsing where a single system is required to be successful with different languages. This research may bring us closer to understanding the linguistic capacity of human brain, and thus to building better NLP systems.

REFERENCES

Barbero, C., Lesmo, L., Lombardo, V. and Merlo, P. (1998). Integration of syntactic and lexical information in a hierarchical dependency grammar. In Kahane, S. and Polguere, A. (eds), *Proceedings of the Workshop on Processing of Dependency-Based Grammars (ACL-COLING)*, pp. 58-67.

Böhmová A., Hajič, J., Hajičová E., and Barbora Hladká . (2003). The Prague Dependency Treebank. In: Anne Abeille (editor). *Treebanks. Building and Using Parsed Corpora*. Kluwer Academic Publishers, pp. 103-127.

Buchholz, S., and Marsi, E. (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing, *Proceedings of the 10th Conference on Natural Language Learning*, pp. 149-164.

Çakıcı, R. and Baldridge, J., 2006. Projective and Non-Projective Turkish Parsing. *Proceedings of the Fifth Conference on Treebanks and Linguistic Theories*, (pp. 19-30).

Çakıcı, R., (2008). (to appear) *Parsing models for a highly inflective language.* PhD Thesis, School of Informatics, University of Edinburgh, UK.

Collins, M., Hajič, J., Brill, E., Ramshaw, L. and Tillmann, C. (1999). A Statistical Parser for Czech. *Proceedings of the 37th Meeting of the Association for Computational Linguistics (ACL)*,(pp. 505-512).

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

Duchier, D. (2001) Lexicalized Syntax and Topology for Non-projective Dependency Grammar. *Proceeding of the Joint Conference on Formal Grammars and Mathematics of Language FGMOL 01*.

Eisner, J. M. (1996). Three New Probabilistic Models for Dependency Parsing: An exploration. *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, (pp. 340-345).

Eisner, J. M. (2000). Bilexical Grammars and Their Cubic-time Parsing Algorithms. In Bunt, H. and Nijholt, A. (eds), *Advances in Probabilistic and Other Parsing Technologies*, Kluwer, (pp. 29-62).

Gaifman, H. (1965). Dependency Systems and Phrase-Structure Systems. *Information and Control* .8: 304-337.

Gerdes, K and Kahane S. (2001). Word order in German: A Formal Dependency Grammar Using a Topological Hierarchy, *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* , Toulouse, France, pp. 220-227.

Gladkij, A.V. and Mel'čuk, I.A. (1975). Tree Grammars: I. A Formalism for Syntactic Transformations in Natural Languages. *Linguistics*, 50:47-82.

Gross., M. (1964). On the equivalence of models of language used in the fields of mechanical translation and information retrieval. *Information Storage and Retrieval*, 2(1):43-57.

Hays, D.G. (1964). Dependency Theory: A Formalism and Some Observations. *Language*, 40(4):511-525.

Hellwig, P. (1986). Dependency Unification Grammar. *Proceedings of the 11th International Conference on Computational Linguistics (COLING)*, pp. 195-198.

Hellwig, P. (2003). Dependency unification grammar. In Agel, V., Eichinger, L.M., Eroms, H.-W., Hellwig, P., Heringer, H. J. and Lobin, H. (eds), *Dependency*

and Valency, Walter de Gruyter, pp. 593-635.

Hudson, R.A. (1990). *English Word Grammar*. Blackwell.

Hudson, R.A. (1984). *Word Grammar*. Blackwell

Joshi, A.K. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In David R. Dowty, Lauri Karttunen, and Arnold M. Z w i c k y, e d i t o r s, *Natural Language Parsing,* (pp. 206-250). Cambridge University Press, Cambridge, UK.

Joshi, A.K., Levy, L.S., and Takahashi, M. (1975). Tree Adjunct Grammars. *Journal Computer Systems Science*, 10(1).

Kruijff, G-J. M. (2001). A Categorial-Modal Architecture of Informativity: Dependency Grammar Logic and Information Structure, Ph.D. thesis, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Kudo, T. and Matsumoto, Y. (2000). Japanese Dependency Structure Analysis Based on Support Vector Machines. P*roceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, pp. 18-25.

Kudo, T. and Matsumoto, Y. (2002). Japanese Dependency Analysis Using Cascaded Chunking. *Proceedings of the Sixth Workshop on Computational Language Learning (CoNLL)*, pp. 63-69.

Marcus, M. P., Santorini, B. and Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19: 313-330.

McDonald, R., Crammer, K. and Pereira, F. (2005 a). Online Large-margin Training of Dependency Parsers. P*roceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 91-98.

McDonald, R., Pereira F., Ribarov K., and Hajič J. (2005 b). Non-projective Dependency Parsing u s - ing Spanning Tree Algorithms. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP),* pp. 523-530,

Mel'čuk, I.A. (1988). *Dependency Syntax: Theory and Practice*. State Univ. of New York Press.

Menzel, W. and Schröder, I. (1998). Decision Procedures for Dependency Parsing Using Graded Constraints. In Kahane, S. and Polguere, A. (eds), *Proceedings of the Workshop on Processing of Dependency-Based Grammars*, pp. 78-87.

Nivre, J., Hall, J. and Nilsson, J. (2004). Memory-based Dependency Parsing. In Ng, H. T. and Riloff,E. (eds), *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL)*, pp. 49-56.

Nivre, J., Hall, J., Kübler, S., McDonald, R., Yüret, D., Nilsson, J. and Riedel S. (2007). The CoNLL 2007 Shared Task on Dependency Parsing. *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pp. 915-932,

Petkevič V. (1987). A New Dependency Based Specification of Underlying Representations of Sentences. *Theoretical Linguistics* 14: 143-172.

Petkevič V. (1995). A new Formal Specification of Underlying Representations. *Theoretical Linguistics* 21: 7-61.

Quirk, C., Menezes, A. and Cherry, C. (2005). Dependency treelet translation: Syntactically informed phrasal SMT. *In 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 271-279.

Riedel, S., Çakıcı, R., and Meza-Ruiz, I. (2006). Multilingual dependency parsing with incremental integer linear programming. *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*.

Robinson, J. J. (1970). Dependency Structures and Transformational Rules. *Language* 46: 259–285.

Starosta, S. (1988). *The Case for Lexicase: An Outline of Lexicase Grammatical Theory*. Pinter Publishers.

Samuelsson, C. (2000). A Statistical Theory of Dependency Syntax. *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*.

Steedman, M. (2000). *The Syntactic Process*, The MIT Press, Cambridge, MA.

D

Sgall,P., Nebeský L., Goralčíková, A. and Hajičová E. (1969). *A Functional Approach to Syntax in Generative Description of Language*, Elsevier, New York.

Sgall, P., Hajičová, E. and Panevová, J. (1986). *The Meaning of the Sentence in Its Pragmatic Aspects*. Reidel.

Tesnière, L. (1959). *Eléments de syntaxe structurale*. Klinsieck, Paris, France.

Tapanainen, P. and Jarvinen, T. (1997). A Non-projective Dependency Parser. *Proceedings of the 5thConference on Applied Natural Language Processing*, pp. 64-71.

Vijay-Shanker,K. and Weir, D. (1994). The Equivalence of Four Extensions of Context-free Grammar. *Mathematical Systems Theory*, 27:511-546.

Yamada, H. and Matsumoto, Y. (2003). Statistical Dependency Analysis with Support Vector Machines. In Van Noord, G. (ed.), *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pp. 195-206.

KEY TERMS

Corpus (corpora plural): A collection of written or spoken material in machine-readable form.

Machine Translation (MT): The act of translating something by means of a machine, especially a computer.

Morpheme: The smallest unit of meaning. A word may consist of one morpheme (need), two morphemes (need/less, need/ing) or more (un/happi/ness).

Phrase Structure Tree: A structural representation of a sentence in the form of an inverted tree, with each node of the tree labelled according to the phrasal constituent it represents.

Rule-Based Parser: A parser that uses hand written (designed) rules as opposed to rules that are derived from the data.

Statistical Parser: A group of parsing methods within NLP. The methods have in common that they associate grammar rules with a probability.

Treebank: A text-corpus in which each sentence is annotated with syntactic structure. Syntactic structure is commonly represented as a tree structure. Treebanks can be used in corpus linguistics for studying syntactic phenomena or in computational linguistics for training or testing parsers.

ENDNOTE

[1]   Some languages are not included in both tasks. The information in the first and second columns of each set belong to CoNLL 2006 and 2007 training data respectively.

Designing Unsupervised Hierarchical Fuzzy Logic Systems

M. Mohammadian
University of Canberra, Australia

INTRODUCTION

Systems such as robotic systems and systems with large input-output data tend to be difficult to model using mathematical techniques. These systems have typically high dimensionality and have degrees of uncertainty in many parameters. Artificial intelligence techniques such as neural networks, fuzzy logic, genetic algorithms and evolutionary algorithms have created new opportunities to solve complex systems. Application of fuzzy logic [Bai, Y., Zhuang H. and Wang, D. (2006)] in particular, to model and solve industrial problems is now wide spread and has universal acceptance. Fuzzy modelling or fuzzy identification has numerous practical applications in control, prediction and inference. It has been found useful when the system is either difficult to predict and or difficult to model by conventional methods. Fuzzy set theory provides a means for representing uncertainties. The underlying power of fuzzy logic is its ability to represent imprecise values in an understandable form. The majority of fuzzy logic systems to date have been static and based upon knowledge derived from imprecise heuristic knowledge of experienced operators, and where applicable also upon physical laws that governs the dynamics of the process.

Although its application to industrial problems has often produced results superior to classical control, the design procedures are limited by the heuristic rules of the system. It is simply assumed that the rules for the system are readily available or can be obtained. This implicit assumption limits the application of fuzzy logic to the cases of the system with a few parameters. The number of parameters of a system could be large. The number of fuzzy rules of a system is directly dependent on these parameters. As the number of parameters increase, the number of fuzzy rules of the system grows exponentially.

Genetic Algorithms can be used as a tool for the generation of fuzzy rules for a fuzzy logic system. This automatic generation of fuzzy rules, via genetic algo-

rithms, can be categorised into two learning techniques, supervised and unsupervised. In this paper unsupervised learning of fuzzy rules of hierarchical and multi-layer fuzzy logic control systems are considered. In unsupervised learning there is no external teacher or critic to oversee the learning process. In other words, there are no specific examples of the function to be learned by the system. Rather, provision is made for a task-independent measure of the quality or representation that the system is required to learn. That is the system learns statistical regularities of the input data and it develops the ability to learn the feature of the input data and thereby create new classes automatically [Mohammadian, M., Nainar, I. and Kingham, M. (1997)].

To perform unsupervised learning, a competitive learning strategy may be used. The individual strings of genetic algorithms compete with each other for the "opportunity" to respond to features contained in the input data. In its simplest form, the system operates in accordance with the strategy that 'the fittest wins and survives'. That is the individual chromosome in a population with greatest fitness 'wins' the competition and gets selected for the genetic algorithms operations (cross-over and mutation). The other individuals in the population then have to compete with fit individual to survive.

The diversity of the learning tasks shown in this paper indicates genetic algorithm's universality for concept learning in unsupervised manner. A hybrid integrated architecture incorporating fuzzy logic and genetic algorithm can generate fuzzy rules for problems requiring supervised or unsupervised learning. In this paper only unsupervised learning of fuzzy logic systems is considered. The learning of fuzzy rules and internal parameters in an unsupervised manner is performed using genetic algorithms. Simulations results have shown that the proposed system is capable of learning the control rules for hierarchical and multi-layer fuzzy logic systems. Application areas considered are, hierarchical control of a network of traffic light control and robotic systems.

A first step in the construction of a fuzzy logic system is to determine which variables are fundamentally important. Any number of these decision variables may appear, but the more that are used, the larger the rule set that must be found. It is known [Raju, S., Zhou J. and Kisner, R. A. (1990), Raju G. V. S. and Zhou, J. (1993), Kingham, M., Mohammadian, M, and Stonier, R. J. (1998)], that the total number of rules in a system is an exponential function of the number of system variables. In order to design a fuzzy system with the required accuracy, the number of rules increases exponentially with the number of input variables and its associated fuzzy sets for the fuzzy logic system. A way to avoid the explosion of fuzzy rule bases in fuzzy logic systems is to consider Hierarchical Fuzzy Logic Control (HFLC) [Raju G. V. S. and Zhou, J. (1993)]. A learning approach based on genetic algorithms [Goldberg, D. (1989)] is discussed in this paper for the determination of the rule bases of hierarchical fuzzy logic systems.

THE GENETIC FUZZY RULE GENERATOR ARCHITECTURE

In this section we show how to learn the fuzzy rules in a fuzzy logic rule base using a genetic algorithm. The full set of fuzzy rules is encoded as a single string in the genetic algorithm population. To facilitate this we develop the genetic fuzzy rule generator whose architecture consists of five basic steps

1. Divide the input and output spaces of the system to be controlled into fuzzy sets (regions),

2. Encode the fuzzy rules into bit-string of 0 and 1,
3. Use a genetic algorithm as a learning procedure to generate set of fuzzy rules,
4. Use a fuzzy logic controller to assess the set of fuzzy rules and assign a value to each generated set of fuzzy rules,
5. Stop generating new sets of fuzzy rules once some performance criteria is met,

Figure 1 shows the genetic fuzzy rule generator architecture graphically. Suppose we wish to produce fuzzy rules for a fuzzy logic control with two inputs and single output. This simple two-input u_1, u_2 single-output y case is chosen in order to clarify the basic ideas of our new approach. Extensions to multi-output cases are straightforward. For more information on multi-output cases refer to Mohammadian et al [Mohammadian, M. and Stonier, R J., (1998)].

As a first step we divide the domain intervals of u_1, u_2 and y into different fuzzy sets. The number of the fuzzy sets is application dependent. Assume that we divide the interval for u_1, u_2 and y into 5, 7 and 7 fuzzy sets respectively. For each fuzzy set we assign a fuzzy membership function. Therefore a maximum of 35 fuzzy rules can be constructed for this system. Now the fuzzy rule base can be formed as a 5×7 table with cells to hold the corresponding actions that must be taken given the condition corresponding to u_1, u_2 are satisfied.

In step 2 we encode the input and output fuzzy sets into bit-strings (of 0 and 1). Each complete bit-string consists of 35 fuzzy rules for this example and each

Figure 1. Genetic fuzzy rule generator architecture
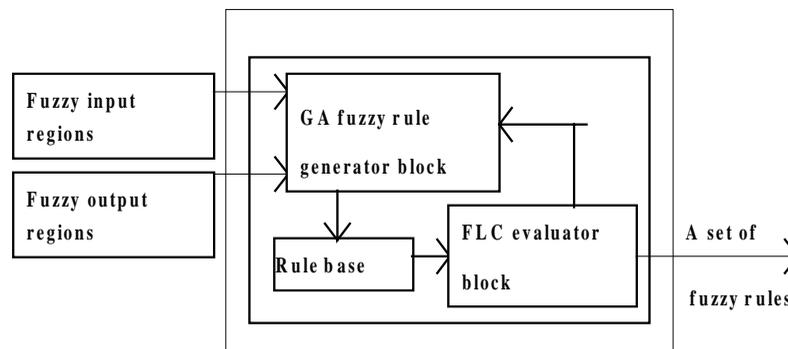
fuzzy rule base has the same input conditions but may have different output control signal assigned to it. Therefore we need only to encode the output signal of the fuzzy rule bit-strings into a complete bit-string. This will save the processing time for encoding and decoding of genetic algorithm's strings. In this case the length of an individual string has been reduced from 665 bits (i.e. 19×35) to 245 bits (i.e. 7×35). The choice of output control signal to be set for each fuzzy rule is made by the genetic algorithm. It initialises randomly a population of complete bit-strings. Each of these bit-strings is then decoded into fuzzy rules and evaluated by fuzzy logic controller to determine the fitness value for that bit-string. Application of proportional selection and mutation and one-point crossover operations can now proceed. Selection and crossover are the same as in simple genetic algorithms while the mutation operation is modified. Crossover and mutation take place based on the probability of crossover and mutation respectively. The mutation operator is changed to suit this problem, namely, an allele is selected at random and it is replaced by a random number ranging from 1

to 7 which represents in this example the five output fuzzy sets. The genetic algorithm process performs a self-directed search according to fitness value. In all applications in this paper we seek to minimise the fitness function. The process can be terminated after a desired number of generations or when the fitness value of the best string in a generation is less than some prescribed level.

VARIABLE SELECTION AND RULE BASE DECOMPOSITION

Traffic Light Control

Traffic light control is widely used to resolve conflicts among vehicle movements at intersections. The control system at each signalised intersection consists of the following three control elements, cycle time, phase splits and offset. Cycle time is the duration of completing all phases of a signal; phase split is the division of the cycle time into periods of green phase for competing approaches; and offset is the time difference in the starting times of the green phases of adjacent intersections.

In [Nainar, I., Mohammadian, M., Stonier, R. J. and Millar, J. (1996)] a fuzzy logic control scheme is proposed to overcome the lack of interactions between the neighbouring intersections. First, a traffic model is developed and a fuzzy control scheme for regulating the traffic flow approaching a single traffic intersection is proposed. A new fuzzy control scheme employing a supervisory fuzzy logic controller is then proposed to coordinate the three intersections based on the traffic conditions at all three intersections. Simulation results established the effectiveness of proposed scheme. Figure 2 shows the three intersections used in the simulation.

A supervisory fuzzy logic control system is then developed to coordinate the three intersections far more effectively than the three local fuzzy logic control systems. This is because using supervisory fuzzy logic controller each intersection is coordinated with all its neighbouring intersections [Nainar, I., Mohammadian, M., Stonier, R. J. and Millar, J. (1996)]. The fuzzy knowledge base of the supervisory fuzzy logic controller was learnt using genetic algorithms. The supervisory fuzzy logic controller developed to coordinate the three intersections coordinated the traffic

Figure 2. Three adjacent intersections
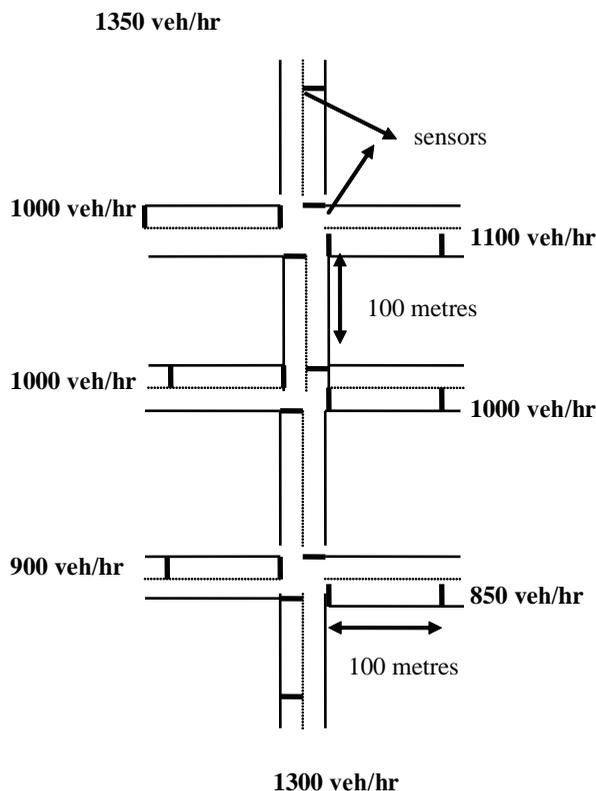
signals far more effectively than the three local fuzzy logic controllers. This is because using supervisory fuzzy logic controller each intersection is coordinated with all its neighbouring intersections. This proposed fuzzy logic control scheme can be effectively applied to on-line traffic control because of its ability to handle extensive traffic situations. Simulations results have shown that the multi-layer fuzzy logic system consisting of three local fuzzy logic controllers and the supervisory fuzzy logic controller is capable of reducing the waiting time on the network of the traffic intersections [Nainar, I., Mohammadian, M., Stonier, R. J. and Millar, J. (1996)].

Collision-Avoidance in a Robot System

Consider the following collision-avoidance problem in a simulated, point mass, two robot system. A three-level hierarchical, fuzzy logic system was proposed to solve the problem, full details can be found in [Mohammadian, M. and Stonier, R. J. (1998)], see also [Mohammadian, M. and Stonier, R. J. (1995)]. In the first layer, two knowledge bases, one for each robot, are developed to find the steering angle to control each robot to its target. In the second layer two new knowledge bases are developed using the knowledge in the first layer to control the speed of each robot so that each robot approaches its target with near zero speed. Finally in the third layer, a single knowledge base is developed to modify the controls of each robot to avoid collision in a restricted common workspace, see Figure 3.

In Figure 3, $x; y$ gives the physical position of the robot on the plane, ϕ is the directional heading of the robot, q is the steering angle, D_1 and D_2 are the distances of the two robots from there respective targets, S_1 and S_2 are the speeds of the two robots, D is the distance between the two robots and q_1', q_2', q_1'', q_2'', s_1' and s_2' are the updates of variable outputs for the last two layers. An important issue in this example is that of learning knowledge in a given layer sufficient for use in higher layers. In the first layer of the hierarchical fuzzy logic system, ignoring the possibility of collision, steering angles for the control of each robot to their associated target were determined by genetic algorithms. In the second layer genetic algorithm was used to determine adjustments to steering angle and speed of each robot to control the speed of the robot when arriving to its target. Next another layer is developed to adjust the speed and steering angle of the robots to avoid collision of the robots. Consider the knowledge base of a single robot in layer one. It is not sufficient to learn a fuzzy knowledge base from an initial configuration and use this knowledge base for information on the steering angle of the robot to learn fuzzy controllers in the second layer. Quite clearly this knowledge base is only guaranteed to be effective from this initial configuration as not all the fuzzy rules will have fired in taking the robot to its target. We have to find a knowledge base that is effective to some acceptable measure, in controlling the robot to its target from 'any' initial configuration. One way is to first learn a set of local fuzzy controllers, each knowledge base learnt by an

Figure 3. Hierarchical structure for collision-avoidance
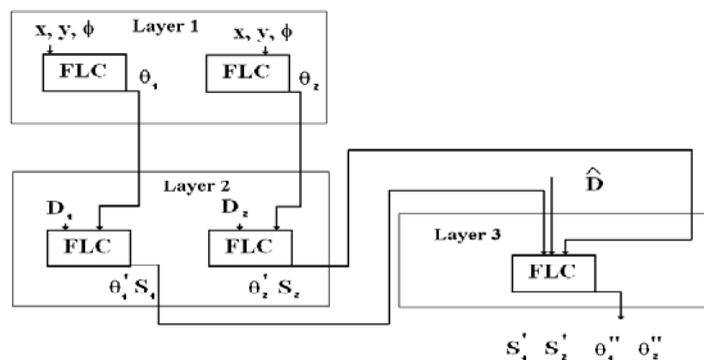
genetic algorithms from a given initial configuration within a set of initial configurations spread uniformly over the configuration space. These knowledge bases can then be *fused* through a *fuzzy amalgamation* process [Stonier, R. J. and Mohammadian, M. (1995)] into the global (final), fuzzy control knowledge base. An alternative approach [Mohammadian, M. and Stonier, R. J. (1996), Stonier, R. J. and Mohammadian, M. (1998)], is to develop an genetic algorithms to learn directly the 'final' knowledge base by itself over the region of initial configurations.

In conclusion the proposed hierarchical fuzzy logic system is capable of controlling the multi-robot system successfully. By using hierarchical fuzzy logic system the number of control laws is reduced. In the first layer of hierarchical fuzzy logic system ignoring the possibility of collision, steering angles for the control of each robot to their associated target were determined by genetic algorithms. In the second layer genetic algorithm was used to determine adjustments to steering angle and speed of each robot to control the speed of the robot when arriving to its target. Next another layer is developed to adjust the speed and steering angle of the robots to avoid collision of the robots. If only one fuzzy logic system was used to solve this problem with the inputs x, y, ϕ of each robot and D each with the same fuzzy sets described in this paper then there would be 153125 fuzzy rule needed for its fuzzy knowledge base. Using a hierarchical fuzzy logic system there is a total number of 1645 fuzzy rules for this system. The hierarchical concept learning using the proposed method makes easier the development of fuzzy logic control systems, by encouraging the development of fuzzy logic controllers where the large number of systems parameters inhibits the construction of such controllers. For more details, we refer the reader to the cited papers.

ISSUES IN RULE BASE IDENTIFICATION

Research into this area has been described as genetic fuzzy systems using the classical genetic algorithm has been surveyed by Cordon [Cordon, O., Herrera, F. and Zwir, I. (2002)], see also [Cordon, O., Herrera, F., Hoffmann, F. and Magdalena, L. (2001)]. Genetic algorithms is employed to learn or tune different components of a fuzzy logic system such as the fuzzy

knowledge base and the membership functions in the inference process. This is usually accomplished by using a genetic algorithms to produce the "best" fuzzy rules and membership functions/parameters with respect to an optimisation criterion. There are three main approaches in the literature for learning the rules in a fuzzy knowledge base. They are, the Pittsburgh approach, the Michigan approach and the iterative rule-learning approach [Cordon, O., Herrera, F., Hoffmann, F. and Magdalena, L. (2001)]. The Pittsburgh and Michigan approaches are the most commonly used methods in the area.

Research by the authors, colleagues and postgraduate students has predominately used the Pittsburgh approach with success in learning the fuzzy rules in complex systems, across hierarchical and multi-layered structures in problems [Stonier, R. J. and Zajaczkowski, J. (2003), Kingham, M., Mohammadian, M, and Stonier, R. J. (1998), Mohammadian, M. and Kingham, M. (2004), Mohammadian, M. (2002), Nainar, I., Mohammadian, M., Stonier, R. J. and Millar, J. (1996), Mohammadian, M. and Stonier, R. J. (1998), Stonier, R. J. and Mohammadian, M. (1995), Thomas, P. J. and Stonier, R. J. (2003), Thomas, P. J. and Stonier, R. J. (2003a)].

FUTURE TRENDS

In using the Pittsburgh approach the coding of the fuzzy rule base as a linear string in an evolutionary algorithm has its drawbacks other than the string may even be relatively large in length under decomposition into multi-layer and hierarchical structures. One is that this is a specific linear encoding of a nonlinear structure and typical one-point crossover when implemented introduces bias when reversing the coding to obtain the fuzzy logic rule base. Using co-evolutionary algorithms is also another option that needs further investigation.

CONCLUSION

This paper described the issues in the construction of a hierarchical fuzzy logic system to model a complex (nonlinear) system. The learning of fuzzy rules in such systems using genetic algorithms was proposed and it was shown to be feasible. Whilst the decomposition

into hierarchical/multi-layered fuzzy logic sub-systems reduces greatly the number of fuzzy rules to be defined and to be learnt, other issues arise such as the decomposition is not unique and that it may give rise to variables with no physical significance. This can raise then major difficulties in obtaining a complete class of rules from experts even when the number of variables is small.

ACKNOWLEDGMENT

REFERENCES

Bai, Y., Zhuang H. and Wang, D. (2006), Advanced Fuzzy Logic Technologies in Industrial Applications, Springer Verlag, USA, ISBN 1-84628-468-6.

Cordon, O., Herrera, F. and Zwir, I. (2002), Linguistic Modeling of Hierarchical Systems of Linguistic Rules, *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 1, 2-20, USA.

Cordon, O., Herrera, F., Hoffmann, F. and Magdalena, L. (2001), *Genetic Fuzzy Systems Evolutionary Tuning and Learning of Fuzzy Knowledge Bases (Advances in Fuzzy Systems—Applications and Theory Vol. 19)*, World Scientific Publishing, USA, ISBN 981-02-4017-1.

Goldberg, D. (1989), Genetic Algorithms in Search, Optimisation and Machine Learning, Addison–Wesley, USA.

Kingham, M., Mohammadian, M, and Stonier, R. J. (1998), Prediction of Interest Rate using Neural Networks and Fuzzy Logic, *Proceedings of ISCA 7th International Conference on Intelligent Systems*, Melun, Paris, France.

Magdalena, L. (1998), Hierarchical Fuzzy Control of a Complex System using Metaknowledge, *Proceedings of the 7th International conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, France.

Mohammadian, M. (2002), Designing customised hierarchical fuzzy systems for modelling and prediction, *Proceedings of the International Conference on sinulated Evolution and Learning (SEAL'02)*, Singapore, ISBN 9810475233.

Mohammadian, M. and Kingham, M. (2004), An adaptive hierarchical fuzzy logic system for modelling of financial systems, *Journal of Intelligent Systems in Accounting, Finance and Management*, Wiley Interscience, Vol. 12, 61-82.

Mohammadian, M. and Kingham, M. (2005), Intelligent Data Analysis, Decision Making and Modelling Adaptive Financial Systems Using Hierarchical Neural Networks, Knowledge-Base Intelligent Information and Engineering Systems, KES2005, Springer Verlag, Australia, ISBN 3540288953.

Mohammadian, M. and Stonier, R. J. (1995), Adaptive Two Layer Fuzzy Logic Control of a Mobile Robot System, *Proceedings of IEEE Conference on Evolutionary Computing*, Perth, Australia.

Mohammadian, M. and Stonier, R. J. (1996), Fuzzy Rule Generation by Genetic Learning for Target Tracking, *Proceedings of the 5th International Intelligent Systems Conference*, Reno, Nevada.

Mohammadian, M. and Stonier, R. J. (1998), Hierarchical Fuzzy Control, *Proceedings of the 7th International conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, France.

Mohammadian, M., Nainar, I. and Kingham, M. (1997), Supervised and Unsupervised Concept Learning by Genetic Algorithms, Second International ICSC Symposium on Fuzzy Logic and Applications ISFL'97, Zurich, Switzerland.

Mohammadian, M. and Stonier, R J., (1998), " *Hierarchical Fuzzy Logic Control* ", The Seventh Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems - IPMU'98, France.

Nainar, I., Mohammadian, M., Stonier, R. J. and Millar, J. (1996), An adaptive fuzzy logic controller for control fo traffic signals, *Proceedings of the 4th International Conference on Control, Automation, Robotics and Computer Vision (ICARCV'96)*, Singapore.

Raju G. V. S. and Zhou, J. (1993), Adaptive Hierarchical Fuzzy Controller, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 4, 973-980, 1993.

Raju, S., Zhou J. and Kisner, R. A. (1990), Fuzzy logic control for steam generator feedwater control, *Proceedings of American Control Conference*, San Diego, CA, USA, ISBN 1491-1493.

Stonier, R. J. and Zajaczkowski, J. (2003), Hierarchical fuzzy controllers for the inverted pendulum, *Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*, Singapore, ISSN 0219-613, PS01-4-03.

Stonier, R. J. (1999), Evolutionary learning of fuzzy logic controllers over a region of initial states, *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, Vol. 2, 2131-2138, 1999.

Stonier, R. J. and Mohammadian, M. (1995), Self Learning Hierarchical Fuzzy Logic Controller in Multi-Robot Systems, *Proceedings of the IEA Conference Control95*, Melbourne Australia.

Stonier, R. J. and Mohammadian, M. (1996), Intelligent Hierarchical Control for Obstacle- Avoidance, *Computational Techniques and Applications - CTAC95*, World Scientific, 733-740, 1996.

Stonier, R. J. and Mohammadian, M. (1998), Knowledge Acquisition for Target Capture, *Proceedings of the International Conference on Evolutionary Computing ICEC'98*, Anchorage, Alaska, USA.

Thomas, P. J. and Stonier, R. J. (2003a), Evolutionary Learning of a 5I2O Fuzzy Controller Including Wheel Lift Constraint, *Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS2003)*, Singapore, Australia, ISSN 0219-613, PS04-4-01.

Thomas, P. J. and Stonier, R. J. (2003), Hierarchical Fuzzy Control in Robot Soccer using Evolving Algorithms, *Proceedings of the International Congress on Evolutionary Computation (CEC2003)*, Canberra, Australia.

KEY TERMS

Fusing Variables: Fusing variables is a method for reducing the number of rules in a fuzzy rule base. The variables are fused (combined) together before input into the inference engine, thereby reducing the number of rules in the knowledge base.

Fuzzy Logic: Fuzzy sets and Fuzzy Logic were introduced in 1965 by Lotfi Zadeh as a new way to represent vagueness in applications. They are a generalisation of sets in conventional set theory. Fuzzy Logic (FL) aims at modelling imprecise models of reasoning, such as common sense reasoning for uncertain complex processes. A system for representing the meaning of lexically imprecise proposition in natural language structure through the proposition being represented as fuzzy constraints on a variable is provided. Fuzzy logic controllers have been applied to many nonlinear control systems successfully. Linguistic rather than crisp numerical rules are used to control the processes.

Fuzzy Rule Base (Fuzzy If-Then rules): Fuzzy If-Then or fuzzy conditional statements are expressions of the form "If **A** Then **B**", where **A** and **B** are labels of fuzzy sets characterised by appropriate membership functions. Due to their concise form, fuzzy If-Then rules are often employed to capture the imprecise modes of reasoning that play an essential role in the human ability to make decision in an environment of uncertainty and imprecision. The set of If-Then rules relate to a fuzzy logic system that are stored together is called a Fuzzy Rule Base.

Genetic Algorithms: Genetic Algorithms (GAs) are algorithms that use operations found in natural genetics to guide their way through a search space and are increasingly being used in the field of optimisation. The robust nature and simple mechanics of genetic algorithms make them inviting tools for search, learning and optimization. Genetic algorithms are based on computational models of fundamental evolutionary processes such as selection, recombination and mutation.

Genetic Algorithms Components: In its simplest form, a genetic algorithm has the following components:

1. *Fitness* **-** A positive measure of utility, called fitness, is determined for individuals in a population. This fitness value is a quantitative measure of how well a given individual compares to others in the population.
2. *Selection* **-** Population individuals are assigned a number of copies in a mating pool that is used to construct a new population. The higher a population individual's fitness, the more copies in the mating pool it receives.
3. *Recombination* **-** Individuals from the mating pool are recombined to form new individuals, called children. A common recombination method is one-point crossover.
4. *Mutation* **-** Each individual is mutated with some small probability $\ll 1.0$. Mutation is a mechanism for maintaining diversity in the population.

Hierarchical Fuzzy Logic Systems: The idea of hierarchical fuzzy logic control systems is to put the input variables into a collection of low-dimensional fuzzy logic control systems, instead of creating a single high dimensional rule base for a fuzzy logic control system. Each low-dimensional fuzzy logic control system constitutes a level in the hierarchical fuzzy logic control system. Hierarchical fuzzy logic control is one approach to avoid rule explosion problem. It has the property that the number of rules needed to construct the fuzzy system increases only linearly with the number of variables in the system

Unsupervised Learning: In unsupervised learning there is no external teacher or critic to oversee the learning process. In other words, there are no specific examples of the function to be learned by the system. Rather, provision is made for a task-independent measure of the quality or representation that the system is required to learn. That is the system learns statistical regularities of the input data and it develops the ability to learn the feature of the input data and thereby create new classes automatically.

D

Developmental Robotics

Max Lungarella
University of Zurich, Switzerland

Gabriel Gómez
University of Zurich, Switzerland

INTRODUCTION

Human intelligence is acquired through a prolonged period of maturation and growth during which a single fertilized egg first turns into an embryo, then grows into a newborn baby, and eventually becomes an adult individual—which, typically before growing old and dying, reproduces. The developmental process is inherently robust and flexible, and biological organisms show an amazing ability during their development to devise adaptive strategies and solutions to cope with environmental changes and guarantee their survival. Because evolution has selected development as the process through which to realize some of the highest known forms of intelligence, it is plausible to assume that development is mechanistically crucial to emulate such intelligence in human-made artifacts.

BACKGROUND

The idea that development might be a good avenue to understand and construct cognition is not new. Already Turing (1950) suggested that using some kind of developmental approach might be a good strategy. In the context of robotics, many of the original ideas can be traced back to embodied artificial intelligence (embodied AI), a movement started by Rodney Brooks at the beginning of the 1980s (Brooks et al., 1998), and the notion of enaction (Varela et al., 1991) according to which cognitive structures emerge from recurrent sensorimotor patterns that enable action to be perceptually guided. Researchers of embodied AI believe that intelligence can only come from the reciprocal interaction across multiple time scales between brain and body of an agent, and its environment. In a sense, throughout life, experience is learned and common sense is acquired, which then supports more complex reasoning. This general bootstrapping of

intelligence has been called "cognitive incrementalism" (Clark, 2001).

DEVELOPMENTAL ROBOTICS

Developmental robotics (also known as epigenetic or ontogenetic robotics) is a highly interdisciplinary subfield of robotics in which ideas from artificial intelligence, developmental psychology, neuroscience, and dynamical systems theory play a pivotal role in motivating the research (Asada *et al.*, 2001; Lungarella *et al.*, 2003; Weng *et al.*, 2001; Zlatev & Balkenius, 2001). Developmental robotics aims to model the development of increasingly complex cognitive processes in natural and artificial systems and to understand how such processes emerge through physical and social interaction. The idea is to realize artificial cognitive systems not by simply programming them to solve a specific task, but rather by initiating and maintaining a developmental process during which the systems interact with their physical environments (i.e. through their bodies or tools), as well as with their social environments (i.e. with people or other robots). Cognition, after all, is the result of a process of self-organization (spontaneous emergence of order) and co-development between a developing organism and its surrounding environment. Although some researchers use simulated environments and computational models (e.g. Mareschal et al., 2007), often robots are employed as testing platforms for theoretical models of the development of cognitive abilities – the rationale being that if a model is instantiated in a system interacting with the real world, a great deal can be learned about its strengths and potential flaws (Fig. 1). Unlike evolutionary robotics which operates on phylogenetic time scales and populations of many individuals, developmental robotics capitalizes on

Figure 1. Developmental robots. (a) iCub (http://www.robotcub.org) (b) Babybot (http://www.liralab.it/babybot/robot.htm) (c) Infanoid (http://univ.nict.go.jp/people/xkozima/infanoid/robot-eng.html#infanoid).
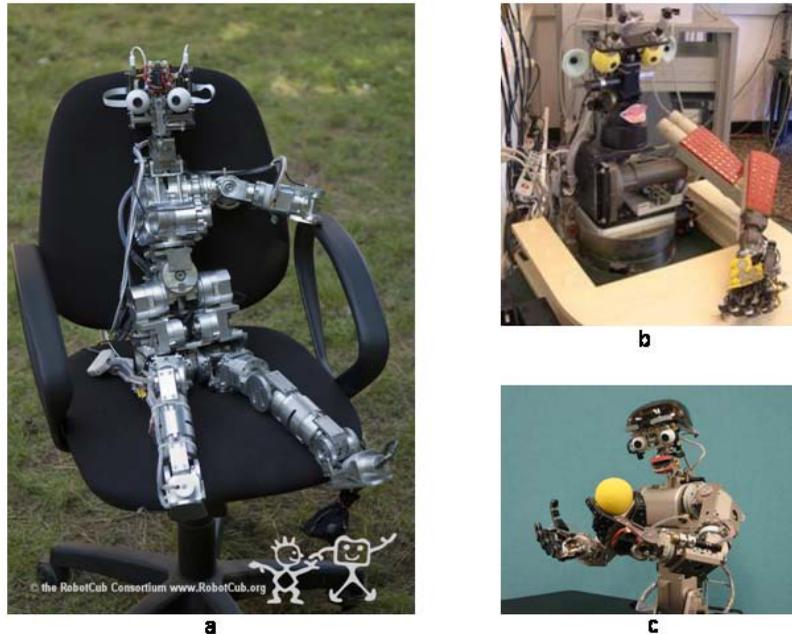
"short" (ontogenetic) time scales and single individuals (or small groups of individuals).

AREAS OF INTEREST

The spectrum of developmental robotics research can be roughly segmented into four primary areas of interest. Although instances may exist that fall into multiple categories, the suggested grouping should provide at least some order in the large spectrum of issues addressed by developmental roboticists.

Socially oriented interaction: This category includes research on robots that communicate or learn particular skills via social interaction with humans or other robots. Examples are imitation learning, communication and language acquisition, attention sharing, turn-taking behavior, and social regulation (Dautenhahn, 2007; Steels, 2006).

Non-social interaction: Studies on robots characterized by a direct and strong coupling between sensorimotor processes and the local environment (e.g. inanimate objects), but which do not interact with other robots or humans. Examples are visually-guided grasping and manipulation, tool-use, perceptual categorization, and navigation (Fitzpatrick *et al.*, 2007; Nabeshima et al., 2006).

Agent-centered sensorimotor control: In these studies, robots are used to investigate the exploration of bodily capabilities, the effect of morphological changes on motor skill acquisition, as well as self-supervised learning schemes not linked to any functional goal. Examples include self-exploration, categorization of motor patterns, motor babbling, and learning to walk or crawl (Demiris & Meltzoff, 2007; Lungarella, 2004).

Mechanisms and principles: This category embraces research on principles, mechanisms or processes thought to increase the adaptivity of a behaving system. Examples are: developmental and neural plasticity, mirror neurons, motivation, freezing and freeing of degrees of freedom, and synergies; characterization of complexity and emergence, study of the effects of adaptation and growth, and practical work on body construction or development (Arbib et al., 2007; Oudeyer et al., 2007; Lungarella & Sporns, 2006).

PRINCIPLES FOR DEVELOPMENTAL SYSTEMS

By contrast to traditional disciplines such as physics or mathematics, which are described by well-known basic principles, the fundamental principles governing the dynamics of developmental systems are unknown. Could there be laws governing developmental systems or a theory? Although various attempts have been initiated (Asada *et al.*, 2001; Brooks *et al.*, 1998; Weng *et al.*, 2001), it is fair to say that to date no such theory has emerged. Here, *en route* to such a theory, we point out a set of candidate principles. An approach based on principles is preferable for constructing intelligent autonomous systems, because it allows capturing design ideas and heuristics in a concise and pertinent way, avoiding blind trial-and-error. Principles can be abstracted from biological systems, and their inspiration can take place at several levels, ranging from a "faithful" replication of biological mechanisms to a rather generic implementation of biological principles leaving room for dynamics intrinsic to artifacts but not found in natural systems. In what follows we summarize five key principles revealed by observations of human development which may be used to construct developmental robots.

The Value Principle

Observations: Value systems are neural structures that mediate value and saliency and are found in virtually all vertebrate species. They are necessary for an organism's behavioral adaptation to salient (meaningful) environmental cues. By linking behavior and neuroplasticity, value systems are essential for deciding what to do in a particular situation (Sporns, 2007).

Lessons for robotics: The action of value systems – through adaptive changes in sensorimotor connections and inputs – enables an embodied agent to learn action strategies without external supervision by increasing the likelihood that a "good" movement pattern can recur in the same behavioral context. Value systems may also be used to guide an exploratory process and hence allow a system to learn sensorimotor patterns more efficiently compared to a pure random or a systematic exploration (Gómez & Eggenberger, 2007). By imposing constraints through value-dependent modulation of saliency, the

search space can be considerably reduced. Examples of value systems in the brain include the dopaminergic, cho-linergic, and noradrenergic systems; based on them, several models have been implemented and embedded in developmental robots (Sporns, 2007).

The Principle of Information Self-Structuring

Observations: Infants frequently engage in repetitive (seemingly dull) behavioral patterns: they look at objects, grasp them, stick them into their mouths, bang them on the floor, and so on. It is through such interactions that intelligence in humans develops as children grow up interacting with their environment (Smith & Breazeal, 2007; Smith & Gasser, 2005).

Lessons for robotics: The first important lesson is that information processing (neural coding) needs to be considered in the context of the embeddedness of the organism within its eco-niche. That is, robots and organisms are exposed to a barrage of sensory data shaped by sensorimotor interactions and morphology (Lungarella & Sporns, 2006). Information is not passively absorbed from the surrounding environment but is selected and shaped by actions on the environment. Second, information structure does not exist before the interaction occurs, but emerges only while the interaction is taking place. The absence of interaction would lead to a large amount of unstructured data and consequently to stronger requirements on neural coding, and – in the worst case – to the inability to learn. It follows that embodied interaction lies at the root of a powerful learning mechanism as it enables the creation of time-locked correlations and the discovery of higher-order regularities that transcend the individual sensory modalities. [Lungarella (2004; "principle of information self-structuring")].

The Principle of Incremental Complexity

Observations: Infants' early experiences are strongly constrained by the immaturity of their sensory, motor, and neural systems. Such early constraints, which at first appear to be an inadequacy, are in fact of advantage, because they effectively decrease the "information overload" that otherwise would overwhelm the infant (Bjorklund & Green, 1992).

Lessons for robotics: In order for an organism – natural or artificial – to learn to control its own

complex brain-body system, it might be a good strategy to start simple and gradually build on top of acquired abilities. The well-timed and gradual co-development of body morphology and neural system provides an incremental approach to deal with a complex and unpredictable world. Early "morphological constraints" and "cognitive limitations" can lead to more adaptive systems as they allow exploiting the role that experience plays in shaping the "cognitive" architecture. If an organism was to begin by using its full complexity, it would never be able to learn anything (Gómez *et al.*, 2004). It follows that designers should not try to "code" a full-fledged ready-to-be-used intelligence module directly into an artificial system. Instead, the system should be able to discover on its own the most effective ways of assembling low-level components into novel solutions [Lungarella (2004; "starting simple"); Pfeifer & Bongard (2007; "incremental process principle")].

The Principle of Interactive Emergence

Observations: Development is not determined by innate mechanisms alone (in other words: not everything should be pre-programmed). Cognitive structure, for instance, is largely dependent on the interaction history of the developing system with the environment in which it is embedded (Hendriks-Jansen, 1996).

Lessons for robotics: In traditional engineering the designer of the system imposes ("hard-wires") the structure of the controller and the controlled system. Designers of adaptive robots, however, should avoid implementing the robot's control structure according to their understanding of the robot's physics, but should endow the robot with means to acquire its own understanding through self-exploration and interaction with the environment. Systems designed for emergence tend to be more adaptive with respect to uncertainties and perturbations. The ability to maintain performance in the face of changes (such as growth or task modifications) is a long-recognized property of living systems. Such robustness is achieved through a host of mechanisms: feedback, modularity, redundancy, structural stability, and plasticity [Dautenhahn (2007; "interactive emergence"); Hendriks-Jansen (1996; "interactive emergence"); Prince et al. (2005; "ongoing emergence")].

The Principle of Cognitive Scaffolding

Observations: Development takes place among conspecifics with similar internal systems and similar external bodies (Smith & Breazeal, 2007). Human infants, for instance, are endowed from an early age with the means to engage in simple, but nevertheless crucial social interactions, e.g. they show preferences for human faces, smell, and speech, and they imitate protruding tongues, smiles, and other facial expressions (Demiris & Meltzoff, 2007).

Lessons for robotics: Social interaction bears many potential advantages for developmental robots: (a) it increases the system's behavioral diversity through mimicry and imitation (Demiris & Meltzoff, 2007); (b) it supports the emergence of language and communication, and symbol grounding (Steels, 2006); and (c) it helps structure the robot's environment by simplifying and speeding up the learning of tasks and the acquisition of skills. Scaffolding is often employed by parents and caretakers (intentionally or not) to support, shape, and guide the development of infants. Similarly, the social world of the robot should be prepared to teach the robot progressively novel and more complex tasks without overwhelming its artificial cognitive structure [Lungarella (2004; "social interaction principle"); Mareschal et al. (2007; "ensocialment"); Smith & Breazeal (2007; "coupling to intelligent others")].

FUTURE TRENDS

The further success of developmental robotics will depend on the extent to which theorists and experimentalists will be able to identify universal principles spanning the multiple levels at which developmental systems operate. Here, we briefly indicate some "hot" issues that need to be tackled *en route* to a theory of developmental systems.

Semiotics: It is necessary to address the issue of how developmental robots (and embodied agents in general) can attribute meaning to symbols and construct semiotic systems. A promising approach, explored under the label of "semiotic dynamics", is that such semiotic systems and the associated information structure are continuously invented and negotiated by groups of people or agents, and are used for communication and information organization (Steels, 2006).

Core knowledge: An organism cannot develop without some built-in ability. If all abilities are built in, however, the organism does not develop either. It will therefore be important to understand with what sort of core knowledge and explorative behaviors a developmental system has to be endowed, so that it can autonomously develop novel skills. One of the greatest challenges will be to identify core abilities and how they interact during development in building basic skills (Spelke, 2000).

Core motives: It is necessary to conduct research on general capacities such as creativity, curiosity, motivations, action selection, and prediction (i.e. the ability to foresee consequence of actions). Ideally, no tasks should be pre-specified to the robot, which should only be provided with an internal abstract reward function and a set of basic motivational (or emotional) drives that could push it to continuously master new know-how and skills (Lewis, 2000; Oudeyer et al., 2007).

Self-exploration: Another important challenge is the one of self-exploration or self-programming (Bongard et al., 2006). Control theory assumes that target values and states are initially provided by the system's designer, whereas in biology, such targets are created and revised continuously by the system itself. Such spontaneous "self-determined evolution" or "autonomous development" is beyond the scope of current control theory and needs to be addressed in future research.

Learning causality: In a natural setting, no teacher can possibly provide a detailed learning signal and sufficient training data. Mechanisms will have to be created to characterize learning in an "ecological context" and for the developing agent to collect relevant learning material on its own. One significant future avenue will be to endow systems with the possibility to recognize progressively longer chains of cause and effect (Chater et al., 2006).

Growth: As mentioned in the introduction, intelligence is acquired through a process of self-assembly, growth, and maturation. It will be important to study how physical growth, change of shape and body composition, as well as material properties of sensors and actuators affect and guide the emergence of cognition. This will allow connecting developmental robotics to computational developmental biology (Gómez & Eggenberger, 2007; Kumar & Bentley, 2003).

CONCLUSION

The study of intelligent systems raises many fundamental, but also very difficult questions. Can machines think or feel? Can they autonomously acquire novel skills? Can the interaction of the body, brain, and environment be exploited to discover novel and creative solutions to problems? Developmental robotics may be an approach to explore such long standing issues. At this point, the field is bubbling with activity. Its popularity is partly due to recent technological advances which have allowed the design of robots whose "kinematic complexity" is comparable to that of humans (Fig. 1). The success of developmental robotics will ultimately depend on whether it will be possible to crystallize its central assumptions into a theory. While much additional work is surely needed to arrive at or even approach a general theory of intelligence, the beginnings of a new synthesis are on the horizon. Perhaps, finally, we will come closer to understanding and building (growing) human-like intelligence. Exciting times are ahead of us.

REFERENCES

Arbib, M., Metta, G., & van der Smagt, P. (2007). Neurorobotics: from vision to action. In: B.Siciliano and O.Khatib (eds.) *Springer Handbook of Robotics* (chapter 63). Springer-Verlag: Berlin.

Asada, M., MacDorman, K.F., Ishiguro, H., & Kuniyoshi, Y. (2001). Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Autonomous Systems*, 37:185–193.

Bjorklund, E.M. & Green, B. (1992). The adaptive nature of cognitive immaturity. *American Psychologist*, 47:46–54.

Bongard, J.C., Zykov, V., & Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314:1118-1121.

Brooks, R.A., Breazeal, C., Irie, R., Kemp, C.C., Marjanovic, M., Scassellati, B., & Williamson, M.M. (1998). Alternative essences of intelligence. *Proc. of 15th Nat. Conf. on Artificial Intelligence*, 961–978.

Chater, N., Tenenbaum, J.B., & Yuille, A. (2006). Probabilistic models of cognition. *Trends in Cognitive Sciences*, 10(7):287–344.

D

Clark, A. (2001). *Mindware: An Introduction to the Philosophy of Cognitive Science*. Oxford University Press: Oxford.

Dautenhahn, K. (2007). Socially intelligent robots: dimensions of human-robot interaction. *Phil. Trans. Roy. Soc. B*, 362:679–704.

Demiris, Y. & Meltzoff, A. (2007). The robot in the crib: a developmental analysis of imitation skills in infants and robots. To appear in: *Infant and Child Development*.

Fitzpatrick, P., Needham, A., Natale, L., & Metta, G. (2007). Shared challenges in object perception for robots and infants. To appear in: *Infant and Child Development*.

Gómez, G., Lungarella, M., Eggenberger, P., Matsushita, K., & Pfeifer, R. (2004). Simulating development in a real robot: on the concurrent increase of sensory, motor, and neural complexity. *Proc. 4th Int. Workshop on Epigenetic Robotics*, pp. 119–122.

Gómez, G. & Eggenberger, P. (2007). Evolutionary synthesis of grasping through self-exploratory movements of a robotic hand. *Proc. IEEE Congress on Evolutionary Computation* (accepted for publication).

Kumar, S. & Bentley, P. (2003). *On Growth, Form and Computers*. Elsevier Academic Press: San Diego, CA.

Hendriks-Jansen, H. (1996). *Catching Ourselves in the Act*. MIT Press: Cambridge, MA.

Lewis, M.D. & Granic, I. (eds.) *Emotion, Development, and Self-Organization – Dynamic Systems Approaches to Emotional Development*. Cambridge University Press: New York.

Lungarella, M. (2004). *Exploring Principles Towards a Developmental Theory of Embodied Artificial Intelligence*. Unpublished PhD Thesis. University of Zurich, Switzerland.

Lungarella, M., Metta, G., Pfeifer, R., & Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, 15:151–190.

Lungarella, M. & Sporns, O. (2006). Mapping information flows in sensorimotor networks. *PLoS Computational Biology*, 2(10):e144.

Mareschal, D., Johnson, M.H., Sirois, S., Spratling, M.W., Thomas, M.S.C., & Westermann, G. (2007). *Neuroconstructivism: How the Brain Constructs Cognition*, Vol.1. Oxford University Press: Oxford, UK.

Nabeshima, C., Lungarella, M., & Kuniyoshi, Y. (2006). Adaptive body schema for robotic tool-use. *Advanced Robotics*, 20(10):1105–1126.

Oudeyer, P.-Y., Kaplan, F., & Hafner, V.V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Trans. Evolutionary Computation*, 11(2):265-286.

Pfeifer, R. & Bongard, J.C. (2007). *How the Body Shapes the Way we Think*. MIT Press: Cambridge, MA.

Prince, C.G., Helder, N.A., & Hollich, G.J. (2005). Ongoing emergence: A core concept in epigenetic robotics. *Proc. 5th Int. Workshop on Epigenetic Robotics*.

Smith, L.B. & Gasser, M. (2005). The development of embodied cognition: Six lessons from babies. *Artificial Life*, 11:13–30.

Smith, L.B. & Breazeal, C. (2007). The dynamic lift of developmental process. *Developmental Science*, 10(1):61–68.

Spelke, E. (2000). Core knowledge. *American Psychologist*, 55:1233–1243.

Sporns, O. (2007). What neuro-robotic models can teach us about neural and cognitive development. In: D. Mareschal et al., (eds.) *Neuroconstructivism: Perspectives and Prospects*, Vol.2, pp. 179–204.

Steels, L. (2006). Semiotic dynamics for embodied agents. *IEEE Intelligent Systems*, 21(3):32–38.

Turing, A.M. (1950). Computing machinery and intelligence. *Mind*, LIX(236):433–460.

Varela, F.J., Thompson, E., & Rosch, E. (1991). *The Embodied Mind*. MIT Press: Cambridge, MA.

Weng, J.J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., & Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*, 291:599–600.

Zlatev, J. & Balkenius, C. (2001). Introduction: why epigenetic robotics? In: C. Balkenius et al., (eds.) *Proc. 1st Int. Workshop on Epigenetic Robotics*, pp. 1–4.

469

KEY TERMS

Adaptation: Refers to particular adjustments that organisms undergo to cope with environmental and morphological changes. In biology one can distinguish four types of adaptation: evolutionary, physiological, sensory, and learning.

Bootstrapping: Designates the process of starting with a minimal set of functions and building increasingly more functionality in a step by step manner on top of structures already present in the system.

Degrees of freedom problem: The problem of learning how to control a system with a very large number of degrees of freedom (also known as Bernstein's problem).

Embodiment: Refers to the fact that intelligence requires a body, and cannot merely exist in the form of an abstract algorithm.

Emergence: A process where phenomena at a certain level arise from interactions at lower levels. The term is sometimes used to denote a property of a system not contained in any one of its parts.

Scaffolding: Encompasses all kinds of external support and aids that simplify the learning of tasks and the acquisition of new skills.

Semiotic Dynamics: Field that studies how meaningful symbolic structures originates, spreads, and evolve over time within populations, by combining linguistics and cognitive science with theoretical tools from complex systems and computer science.

Device–Level Majority von Neumann Multiplexing

Valeriu Beiu
United Arab Emirates University, UAE

Walid Ibrahim
United Arab Emirates University, UAE

Sanja Lazarova-Molnar
United Arab Emirates University, UAE

INTRODUCTION

This chapter starts from an exact gate-level reliability analysis of von Neumann multiplexing using majority gates of increasing fan-ins ($\Delta = 3, 5, 7, 9, 11$) at the smallest redundancy factors ($R_F = 2\Delta$), and details an accurate device-level analysis. The analysis complements well-known theoretical and simulation results. The gate-level analysis is exact as obtained using exhaustive counting. The extension (of the exact gate-level analysis) to device-level errors will allow us to analyze von Neumann majority multiplexing with respect to device malfunctions. These results explain abnormal behaviors of von Neumann multiplexing reported based on Monte Carlo simulations. These analyses show that *device-level reliability results are quite different from the gate-level ones, and could have profound implications for future (nano)circuit designs.*

SIA (2005) predicts that the semiconductor industry will continue its success in scaling CMOS for a few more generations. This scaling should become very difficult when approaching 16 nm. Scaling might continue further, but alternative nanodevices might be integrated with CMOS on the same platform. Besides the higher sensitivities of future ultra-small devices, the simultaneous increase of their numbers will create the ripe conditions for an inflection point in the way we deal with reliability.

With geometries shrinking the *available reliability margins* of the future nano(devices) are considerably being reduced (Constantinescu, 2003), (Beiu et al., 2004). From the chip designers' perspective, reliability currently manifests itself as time-dependent uncertainties and variations of electrical parameters. In the nano-era, these *device-level parametric uncertainties*

are becoming too high to handle with prevailing worst-case design techniques—without incurring significant penalty in terms of area, delay, and power/energy. The global picture is that reliability looks like one of the greatest threats to the design of future ICs. For emerging nanodevices and their associated interconnects the anticipated probabilities of failures, could make future nano-ICs prohibitively unreliable. The present design approach based on the conventional zero-defect foundation is seriously being challenged. Therefore, fault- and defect-tolerance techniques will have to be considered from the early design phases.

Reliability for beyond CMOS technologies (Hutchby et al., 2002) (Waser, 2005) is expected to get even worse, as device failure rates are predicted to be as high as 10% for single electron technology, or SET (Likharev, 1999), going up to 30% for self-assembled DNA (Feldkamp & Niemeyer, 2006) (Lin et al., 2006). Additionally, a comprehensive analysis of carbon nano tubes for future interconnects (Massoud & Nieuwoudt, 2006) estimated the variations in delay at about 60% from the nominal value. Recently, defect rates of 60% were reported for a 160 Kbit molecular electronic memory (Green et al., 2007). Achieving 100% correctness with 10^{12} nanodevices will be not only outrageously expensive, but plainly impossible! Relaxing the requirement of 100% correctness should reduce *manufacturing*, *verification*, and *test* costs, while leading to more transient and permanent errors. It follows that most (if not all) of these errors will have to be compensated by architectural techniques (Nikolić et al., 2001) (Constantinescu, 2003) (Beiu et al., 2004) (Beiu & Rückert, 2009).

From the system design perspective errors fall into: *permanent* (defects), *intermittent*, and *transient*

(faults). The origins of these errors can be found in the manufacturing process, the physical changes appearing during operation, as well as sensitivity to internal and external noises and variations. It is not clear if emerging nanotechnologies will not require new fault models, or if multiple errors might have to be dealt with. Kuo (2006) even mentioned that: *"we are unsure as to whether much of the knowledge that is based on past technologies is still valid for reliability analysis."* The well-known approach for fighting against errors is to incorporate redundancy: either *static* (in space, time, or information) or *dynamic* (requiring fault detection, location, containment, and recovery). Space (hardware) redundancy relies on voters (generic, inexact, mid-value, median, weighted average, analog, hybrid, etc.) and includes: modular redundancy, cascaded modular redundancy, and multiplexing like von Neumann multiplexing vN-MUX (von Neumann, 1952), enhanced vN-MUX (Roy & Beiu, 2004), and parallel restitution (Sadek et al., 2004). Time redundancy is trading space for time, while information redundancy is based on error detection and error correction codes.

This chapter explores the performance of vN-MUX when using majority gates of *fan-in* Δ (MAJ-Δ). The aim is to get a clear understanding of the trade-offs between the reliability enhancements obtained when using MAJ-Δ vN-MUX at the smallest redundancy factors $R_F = 2\Delta$ (see Fig. 1) on one side, versus both the fan-ins and the unreliable nanodevices on the other side. We shall start by reviewing some theoretical and simulation results for vN-MUX in Background section.

Exact gate-level simulations (as based on an exhaustive counting algorithm) and accurate device-level estimates, including details of the effects played by nanodevices on MAJ-Δ vN-MUX, are introduced in the Main Focus of the Chapter section. Finally, implications and future trends are discussed in Future Trends, and conclusions and further directions of research are ending this chapter.

BACKGROUND

Multiplexing was introduced by von Neumann as a scheme for reliable computations (von Neumann, 1952). vN-MUX is based on successive computing stages alternating with random interconnection stages. Each computing stage contains a set of redundant gates. Although vN-MUX was originally exemplified for NAND-2 it can be implemented using any type of gate, and could be applied to any level of abstraction (subcircuits, gates, or devices). The 'multiplexing' of each computation tries to reduce the likelihood of errors propagating further, by selecting the more-likely result(s) at each stage. Redundancy is quantified by a redundancy factor R_F, which indicates the multiplicative increase in the number of gates (subcircuits, or devices). In his original study, von Neumann (1952) assumed independent (un-correlated) gate failures pf_{GATE} and very large R_F. The performance of NAND-2 vN-MUX was compared with other fault-tolerant techniques in (Forshaw et al., 2001), and it was analyzed at lower

Figure 1. Minimum redundancy MAJ-Δ vN-MUX: (a) MAJ-3 ($R_F = 6$); and (b) MAJ-5 ($R_F = 10$)

(a)   *(b)*
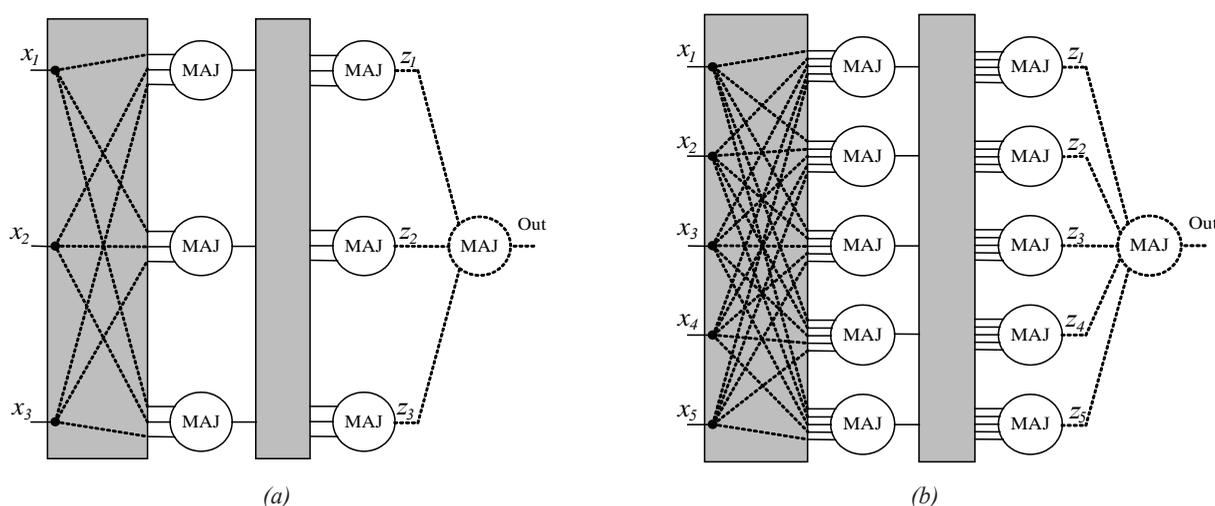
R_F (30 to 3,000) in (Han & Jonker, 2002), while the first exact analysis at very low R_F (3 to 100) for MAJ-3 vN-MUX was done in (Roy & Beiu, 2004).

The issue of which gate should one use is debatable (Ibrahim & Beiu, 2007). It was proven that using MAJ-3 could lead to improved vN-MUX computations only for $pf_{MAJ-3} < 0.0197$ (von Neumann, 1952), (Roy & Beiu, 2004). This outperforms the NAND-2 error threshold $pf_{NAND-2} < 0.0107$ (von Neumann, 1952), (Sadek et al., 2004). Several other studies have shown that the error thresholds of MAJ are higher than those of NAND when used in vN-MUX. Evans (1994) proved that:

$$pf_{MAJ-\Delta} \le \frac{1}{2} - \frac{2^{\Delta-2}}{\Delta \times \mathbf{C}_{\Delta-1}^{(\Delta-1)/2}}, \qquad (1)$$

while the error threshold for NAND-Δ was determined in (Gao et al., 2005) by solving:

$$\left(1+\frac{1}{\Delta}\right) \times \left[\frac{1}{\Delta(1-2pf_{NAND-\Delta})}\right]^{1/(\Delta-1)} = 1 - pf_{NAND-\Delta}. \qquad (2)$$

An approach for getting a better understanding of vN-MUX at very small R_F is to use Monte Carlo simulations (Beiu, 2005), (Beiu & Sulieman, 2006), (Beiu et al., 2006). These have revealed that the reliability of NAND-2 vN-MUX is in fact better than that of MAJ-3 vN-MUX (at $R_F = 6$) for small geometrical variations ν. As opposed to the theoretical results—where the reliability of MAJ-3 vN-MUX is *always* better than NAND-2 vN-MUX—the Monte Carlo simulations showed that MAJ-3 vN-MUX is better than NAND-2 vN-MUX, but only for $\nu > 3.4\%$. Such results were neither predicted (by theory) nor suggested by (gate-level) simulations.

It is to be highlighted here that all the theoretical publications discuss unreliable *organs*, *gates*, *nodes*, *circuits*, or *formulas*, but very few mention devices. For getting a clear picture we have started by developing an exhaustive counting algorithm which exactly calculates the reliability of MAJ-Δ vN-MUX (Beiu et al., 2007). The probability of failure of MAJ-Δ vN-MUX is:

$$Pf_{MAJ-\Delta \text{ vN-MUX}}$$

$$= \sum_{k=0}^{2\Delta} \left[\sum_{i=1}^{\mathbf{C}_{2\Delta}^{k}} \frac{(\#\text{vNfaults})_{ik} \cdot pf_{MAJ-\Delta}^{k} \cdot (1 - pf_{MAJ-\Delta})^{2\Delta-k}}{2^{\Delta}}\right]. (3)$$

The results based on exhaustive counting when varying $pf_{MAJ-\Delta}$ have confirmed both the theoretical and the simulation ones. MAJ-Δ vN-MUX at the minimum redundancy factor $R_F = 2\Delta$ improves the reliability over MAJ-Δ when $pf_{MAJ-\Delta} \le 10\%$, and increasing Δ increases the reliability. When $pf_{MAJ-\Delta}$ 10%, using vN-MUX increases the reliability over that of MAJ-Δ as long as $pf_{MAJ-\Delta}$ is lower than a certain error threshold. If $pf_{MAJ-\Delta}$ is above the error threshold, the use of vN-MUX is detrimental, as the reliability of the system is lower than that of MAJ-Δ. Still, these do not explain the Monte Carlo simulation results mentioned earlier.

MAIN FOCUS OF THE CHAPTER

Both the original vN-MUX study and the subsequent theoretical ones have considered unreliable gates. They did not consider the elementary devices, and assumed that the gates have a fixed (bounding) pf_{GATE}. This assumption ignores the fact that *different gates are built using different (numbers of) devices, logic styles, or (novel) technological principles*. While a standard CMOS inverter has 2 transistors, NAND-2 and MAJ-3 have 4 and respectively 10 transistors. Forshaw et al. (2001) suggested that pf_{GATE} could be estimated as:

$$pf_{GATE} = 1 - (1 - \varepsilon)^n, \qquad (4)$$

where ε denotes the probability of failure of a nanodevice (*e.g.*, transistor, junction, capacitor, molecule, quantum dot, etc.), and n is the number of nanodevices a gate has.

Using eq. 4 as $pf_{MAJ-\Delta} = 1 - (1 - \varepsilon)^{2\Delta}$ the reliabilities have been estimated by modifying the exact counting results reported in (Beiu et al., 2007). The device-level estimates can be seen in Fig. 2. They show that increasing Δ will not necessarily increase the reliability of MAJ-Δ vN-MUX (over MAJ-Δ). This is happening because MAJ-Δ with larger Δ require more nanodevices. In particular, while MAJ-11 vN-MUX is the best solution for $\varepsilon \le 1\text{‰}$ (Fig. 2(a)), it becomes the worst for $\varepsilon > 2\%$ (Fig. 2(b)). Hence, larger fan-ins are advantageous for lower ε ($\le 1\text{‰}$), while small fan-ins perform better for larger ε ($> 1\%$). Obviously, there is a "swapping" region where the ranking is being reversed. We detail Fig. 2(b) for $\varepsilon > 1\%$ (Fig. 2(c)) and for the "swapping" region $1\text{‰} < \varepsilon < 2\%$ (Fig. 2(d), where '○' marks show the envelope). These results imply that increasing Δ and/or

Fig. 2. Probability of failure of MAJ-Δ vN-MUX plotted versus the probability of failure of the elementary (nano)device ε: (a) small ε (< 1‰); (b) large ε (> 1%); (c) detail for ε in between 1% and 10%; (d) detailed view of the "swapping" region (ε in between 1‰ and 2%)

(a)

(b)

(c)

(d)
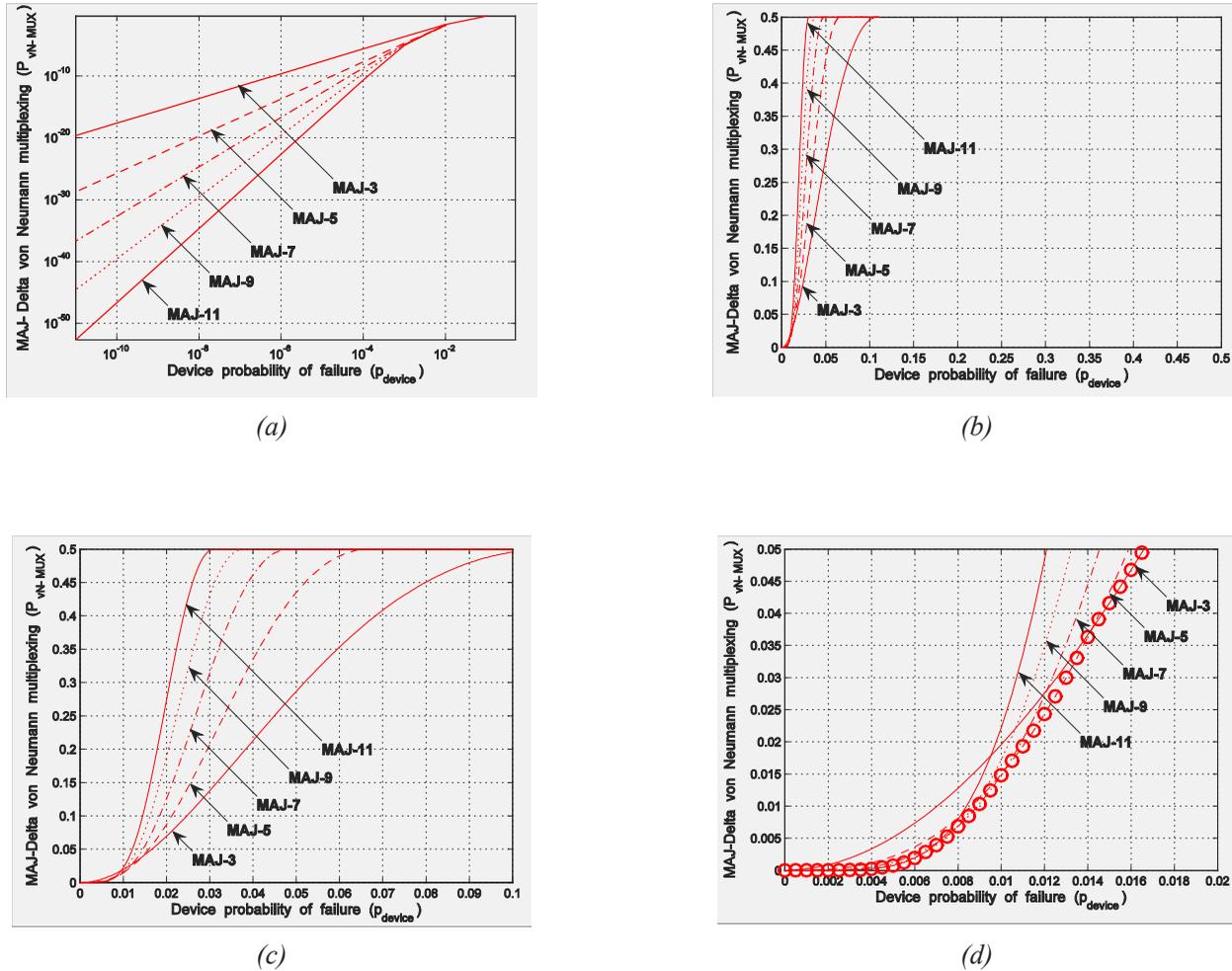
R_F does not necessarily improve the overall system's reliability. This is because increasing Δ and/or R_F leads to increasing the number of nanodevices:

$$N_{\text{MAJ-Δ vN-MUX}} = R_{\text{MAJ-Δ vN-MUX}} \times n_{\text{MAJ-Δ}} = 2Δ \times 2Δ = 4Δ^2. \quad (5)$$

This quadratic dependence on Δ has to be accounted for. Basically, it is ε and the number of devices *N*, and not (only) R_F and pf_{GATE}, which should be used when trying to accurately predict the advantages of vN-MUX—or of any other redundancy scheme.

The next step we took was to compare the device-level estimates of MAJ-Δ vN-MUX (Fig. 3(a)), with the Monte Carlo simulation ones (Fig. 3(b), adapted from (Beiu, 2005) (Beiu & Sulieman, 2006)). The two plots in Fig. 3 have the same vertical scale and exhibit similar shapes. Still, a direct comparison is not trivial as these are mapped against different variables (ε and respectively ν). This similarity makes us confident that the estimated results are accurate and supporting the claim that a simple estimate for pf_{GATE} leads to good approximations at the system level. For other insights the interested reader should consult (Anghel & Nicolaidis, 2007) and (Martorell et al., 2007).

Figure 3. Probability of failure of MAJ-3 vN-MUX: (a) using device-level estimates and exact counting results; (b) C-SET Monte Carlo simulations for MAJ-3 vN-MUX
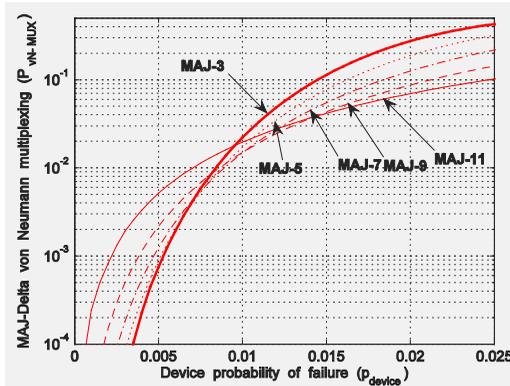
(a)
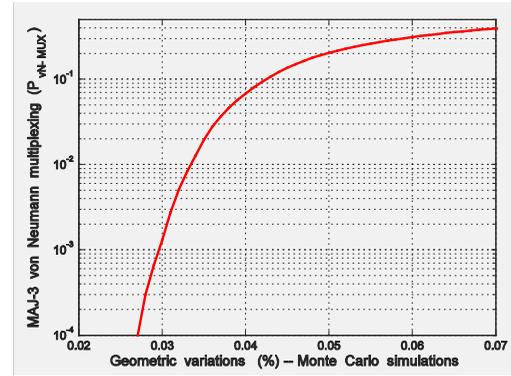
(b)

FUTURE TRENDS

In a first set of experiments, we compared the reliability of MAJ-Δ with the reliability of MAJ-Δ vN-MUX at $R_F = 2\Delta$. For device-level analyses this is not obvious anymore as MAJ-Δ are not on a 45° line, which makes it hard to understand where and by how much vN-MUX improves over MAJ-Δ. The results of these simulations can be seen in Fig. 4, where we have used the same interval $\varepsilon \in [0, 0.11]$ on the horizontal axis. Here again it looks like the smallest fan-in is the best.

A second set of experiments has studied the effect of changing Δ on the error threshold of MAJ-Δ vN-MUX. Fig. 5(a) shows the theoretical gate-level error thresholds (using eq. (1)), as well as the achievable gate-level error thresholds evaluated based on simulations using the exhaustive counting algorithm. Fig. 5(a) shows that the exact gate-level error thresholds are higher than the theoretical gate-level error threshold values (by about 33%). *It would appear that one could always enhance reliability by going for a larger fan-in*. The extension to device-level estimates can be seen in Fig. 5(b), which reveals a completely different picture. These results imply that:

• device-level error thresholds are about 10× less than the gate-level error thresholds;

• device-level error thresholds are decreasing with increasing to fan-ins (exactly the opposite of gate-level error thresholds);

• for vN-MUX, the highest device-level error threshold of about 4% is achieved when using MAJ-3.

CONCLUSION

This chapter has presented a detailed analysis of MAJ-Δ vN-MUX for very small fan-ins: exact for the gate-level and estimated but accurate for the device-level. The main conclusions are as follows.

• Exact gate-level error thresholds for MAJ-Δ vN-MUX are about 33% better than the theoretical ones and increase with increasing fan-in.

• Estimated device-level error thresholds are about 10× lower than gate-level error thresholds and are decreasing with increasing fan-ins—making smaller fan-ins better (Beiu & Makaruk, 1998), (Ibrahim & Beiu, 2007).

• The abnormal (nonlinear) behavior of vN-MUX (Beiu, 2005), (Beiu & Sulieman, 2006) is due to the fact that the elementary gates are made

Figure 4. Probability of failure of MAJ-Δ and MAJ-Δ vN-MUX plotted versus the device probability of failure ε: (a) Δ = 3; (b) Δ = 5; (c) Δ = 7; (d) Δ = 9. For uniformity, the same interval ε ∈ [0, 0.11] was used for all the plots
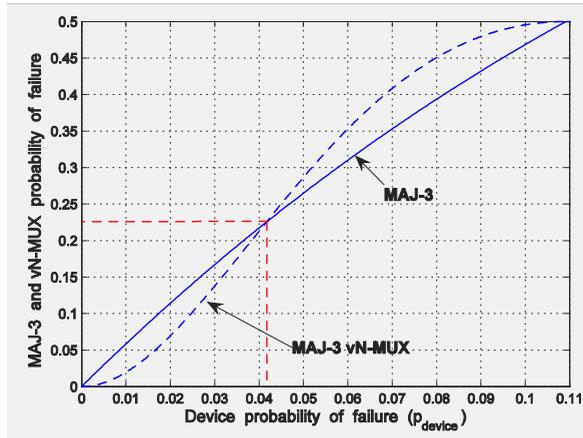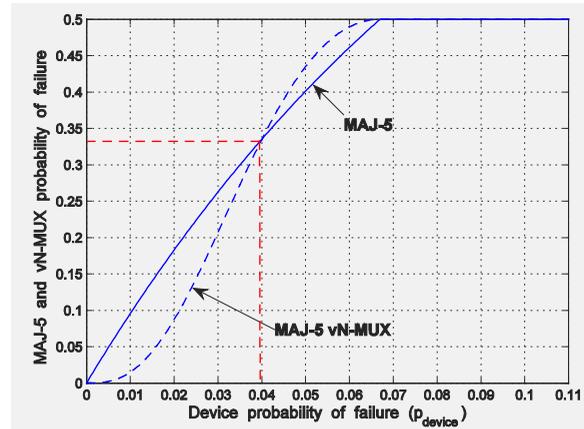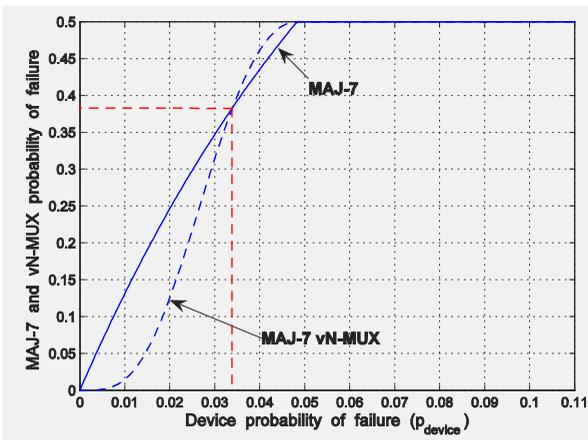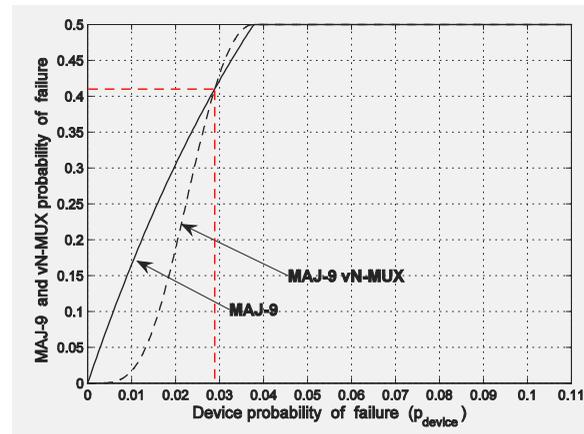
of unreliable nanodevices (implicitly accounted for by Monte Carlo simulations, but neglected by theoretical approaches and gate-level simulations).

• Extending the exact gate-level simulations to device-level estimates using $1 - (1 - \varepsilon)^n$, leads to quite accurate approximations (as compared to Monte Carlo simulations).

• Device-level estimates show much more complex behaviors than those revealed by gate-level analyses (nonlinear for large ε, leading to multiple and intricate crossings).

• Device-level estimates suggest that reliability optimizations for large ε will be more difficult than what was expected from gate-level analyses.

• One way to maximize reliability when ε is large and unknown (*e.g.*, time varying (Srinivasan

Figure 5. Error thresholds for MAJ-Δ vN-MUX versus fan-in Δ: (a) gate-level error threshold, both theoretical (red) and exact (yellow) as obtained through exhaustive counting; (b) device-level error threshold

D
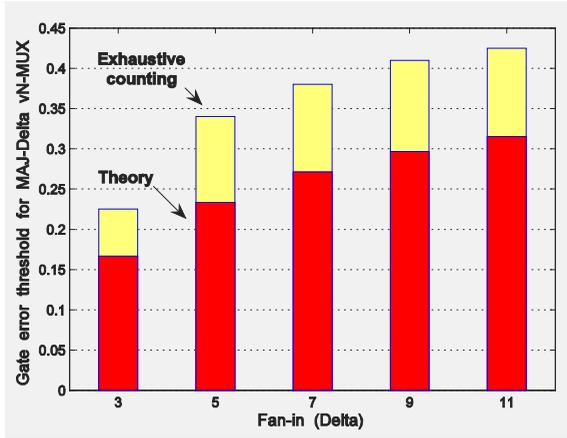
(a)
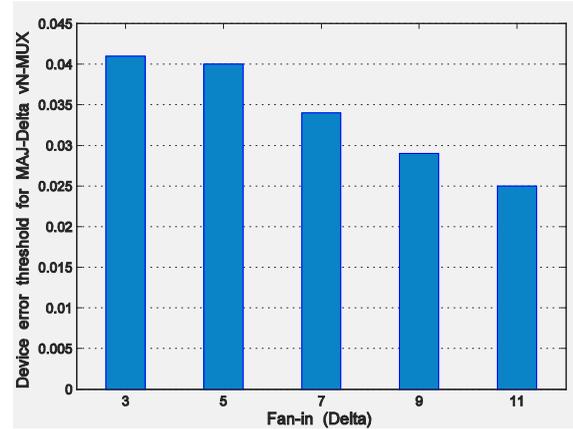
(b)

et al., 2005)) is to rely on 'adaptive' gates, so neural-inspiration should play a(n important) role in future nano-IC designs (Beiu & Ibrahim, 2007).

Finally, precision is very important as "*small errors ... have a huge impact in estimating the required level of redundancy for achieving a specified/target reliability*" (Roelke et al., 2007). It seems that the current gate-level models tend to underestimate reliability, while we do not do a good job at the device-level, with Monte Carlo simulation the only widely used method. More precise estimates than the ones presented in this chapter are possible using Monte Carlo simulations in combination with gate-level reliability algorithms. These are clearly needed and have just started to be investigated (Lazarova-Molnar et al., 2007).

REFERENCES

Anghel, L., & Nicolaidis, M. (2007). Defects Tolerant Logic Gates for Unreliable Future Nanotechnologies. *Proceedings IWANN'07,* San Sebastián, Spain, Springer-Verlag, LNCS 4507, 422-429.

Beiu, V. (2003). Neural Inspired Architectures for Nanoelectronics: Highly Reliable, Ultra Low-power, Reconfigurable, Asynchronous. *Neural Information Processing Systems NIPS'03*, Whistler, Canada, www.eecs.wsu.edu/~vbeiu/workshop_nips03/.

Beiu, V. (2005). The Quest for Practical Redundant Computations. *NSF Workshop on Architectures for Silicon Nanoelectronics & Beyond*, Portland State University, Portland, OR, USA, web.cecs.pdx.edu/~strom/beiu.pdf. [Beiu, V. (2005). The Quest for Reliable Nano Computations. *Proceedings ICM'05*, Islamabad, Pakistan, xix.]

Beiu, V., & Makaruk, H. E. (1998). Deeper Sparsely Nets Can Be Optimal. *Neural Processing Letters*, (8) 3, 201-210.

Beiu, V., & Sulieman, M. H. (2006). On Practical Multiplexing Issues. *Proceedings IEEE-NANO'06*, Cincinnati, USA, 310-313.

Beiu, V., & Ibrahim, W. (2007). On Computing Nano-architectures Using Unreliable Nano-devices. In Lyshevski, S.E. (Editor), *Nano and Molecular Electronics Handbook*, Taylor & Francis, Chapter 12, 1-49.

Beiu, V., & Rückert, U. (Editors) (2009). *Emerging Brain Inspired Nano Architectures*. World Scientific Press.

Beiu, V., Ibrahim, W., & Lazarova-Molnar, S. (2007). What von Neumann Did Not Say About Multiplexing. *Proceedings IWANN'07,* San Sebastián, Spain, Springer-Verlag, LNCS 4507, 487-496.

Beiu, V., Rückert, U., Roy, S., & Nyathi, J. (2004). On Nanoelectronic Architectural Challenges and Solutions. *Proceedings IEEE-NANO'04*, Munich, Germany, 628-631.

Beiu, V., Ibrahim, W., Alkhawwar, Y. A., & Sulieman, M. H. (2006). Gate Failures Effectively Shape Multiplexing. *Proceedings DFT'06*, Washington, USA, 29-35.

Constantinescu, C. (2003). Trends and Challenges in VLSI Circuit Reliability. *IEEE Micro*, (23) 4, 14-19.

Evans, W. S. (1994). Information Theory and Noisy Computation. *PhD dissertation*. *Tech. Rep. TR-94-057*, International Computer Science Institute (ICSI), Berkeley, CA, USA, ftp://ftp.icsi.berkeley.edu/pub/techreports/1994/tr-94-057.pdf. [Evans, W. S., & Schulman, L. J. (2003). On the Maximum Tolerable Noise of k-input Gates for Reliable Computations by Formulas. *IEEE Transactions on Information Theory*, (49) 11, 3094-3098.]

Feldkamp, U., & Niemeyer, C. M. (2006). Rational Design of DNA Nanoarchitectures. *Angewandte Chemie International Edition*, (45) 12, 1856-1876.

Forshaw, M., Nikolić, K., & Sadek, A. S. (2001). ANSWERS: Autonomous Nanoelectronic Systems With Extended Replication and Signaling. *MEL-ARI #28667 Report*, ipga.phys.ucl.ac.uk/research/answers/reports/3rd_year_U CL.pdf.

Gao, J. B., Qi, Y., & Fortes, J.A.B. (2005). Bifurcations and Fundamental Error Bounds for Fault-Tolerant Computations. *IEEE Transactions on Nanotechnology*, (4) 4, 395-402.

Green, J. E., Choi, J. W., Boukai, A., Bunimovich, Y., Johnston-Halperin, E., DeIonno, E., Luo, Y., Sheriff, B. A., Xu, K., Shin, Y. S., Tseng, H.-R., Stoddart, J. F., & Heath, J. R. (2007). A 160-Kilobit Molecular Electronic Memory Patterned at 10^{11} Bits per Square Centimeter. *Nature*, (445) 7126, 414-417.

Han, J., & Jonker, P. (2002). A System Architecture Solution for Unreliable Nanoelectronic Devices," *IEEE Transactions on Nanotechnology*, (1) 4, 201-208.

Hutchby, J. A., Bourianoff, G. I., Zhirnov, V. V., & Brewer, J.E. (2002). Extending the Road beyond CMOS. *IEEE Circuit & Device Magazine*, (18) 2, 28-41.

Ibrahim, W., & Beiu, V. (2007). Long Live Small Fan-in Majority Gates Their Reign Looks Like Coming! *Proceedings ASAP'07*, Montréal, Canada (pp. 278-283).

Kuo, W. (2006). Challenges Related to Reliability in Nano Electronics. *IEEE Transactions on Reliability*, (55) 4, 569-570 [corrections in *IEEE Transactions on Reliability*, (56) 1, 169].

Lazarova-Molnar, S., Beiu, V., & Ibrahim, W. (2007). Reliability: The Fourth Optimization Pillar of Nanoelectronics. *Proc. ICSPC'07*, Dubai, UAE (pp. 73-76).

Likharev, K. K. (1999). Single-Electron Devices and Their Applications. *Proceedings of the IEEE*, (87) 4, 606-632.

Lin, C., Liu, Y., Rinker, S., & Yan, H. (2006). DNA Tile Based Self-Assembly: Building Complex Nanoarchitectures. *ChemPhysChem*, (7) 8, 1641-1647.

Martorell, F., Coțofană, S.D., & Rubio, A. (2007). Fault Tolerant Structures for Nanaoscale Gates. *Proc. IEEE-NANO'07*, Hong Kong, 605-610.

Massoud, Y., & Nieuwoudt, A. (2006). Modeling and Design Challenges and Solutions for Carbon Nanotube-Based Interconnect in Future High Performance Integrated Circuits. *ACM Journal on Emerging Technologies in Computing Systems*, (2) 3, 155-196.

Nikolić, K., Sadek, A. S., & Forshaw, M. (2001). Architectures for Reliable Computing with Unreliable Nanodevices. *Proceedings IEEE-NANO'01*, Maui, USA, 254-259. [Forshaw, M., Crawley, D., Jonker, P., Han, J., & Sotomayor Torres, C. (2004). A Review of the Status of Research and Training into Architectures for Nanoelectronic and Nanophotonic Systems in the European Research Area. *FP6/2002/IST/1*, *#507519 Report*, www.ph.tn.tudelft.nl/People/albert/papers/NanoArchRev_finalV2.pdf.]

Roelke, G., Baldwin, R., & Bulutoglu, D. (2007). Analytical Models for the Performance of von Neumann Multiplexing. *IEEE Transactions on Nanotechnology*, (6) 1, 75-89.

Roy, S., & Beiu, V. (2004). Multiplexing Schemes for Cost-effective Fault-tolerance. *Proceedings IEEE-NANO'04*, Munich, Germany, 589-592. [Roy, S., & Beiu, V. (2005). Majority Multiplexing—Economical Redundant Fault-tolerant Designs for Nanoarchitectures. *IEEE Transactions on Nanotechnology*, (4) 4, 441-451.]

Sadek, A. S., Nikolić, K., & Forshaw, M. (2004). Parallel Information and Computation with Restitution for Noise-tolerant Nanoscale Logic Networks. *Nanotechnology*, (15) 1, 192-210.

SIA – Semiconductor Industry Association (2005). *International Technology Roadmap for Semiconductors*. SEMATECH, USA, public.itrs.net.

Srinivasan, J., Adve, S. V., Bose, P., & Rivers, J. A. (2005). Lifetime Reliability: Toward an Architectural Solution. *IEEE Micro*, (25) 3, 70-80

von Neumann, J. (1952). Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components. *Lecture Notes*, Caltech, Pasadena, CA, USA. [In Shannon, C. E., & McCarthy, J. (Editors) (1956). *Automata Studies*. Princeton University Press, 43-98.]

Waser, R. (Editor) (2005). *Nanoelectronics and Information Technology* (2nd edition). Wiley-VCH.

KEY TERMS

Circuit: Network of devices.

Counting (Exhaustive): The mathematical action of repeated addition (exhaustive considers all possible combinations).

Device: Any physical entity deliberately affecting the information carrying particle (or their associated fields) in a desired manner, consistent with the intended function of the circuit.

Error Threshold: The probability of failure of a component (gate, device) above which the multiplexed scheme is not able to improve over the component itself.

Fan-In: Number of inputs (to a gate).

Fault-Tolerant: The ability of a system (circuit) to continue to operate rather than failing completely (possibly at a reduced performance level) in the event of the failure of some of its components.

Gate (Logic): Functional building block (in digital logic a gate performs a logical operation on its logic inputs).

Majority (Gate): A logic gate of odd fan-in which outputs a logic value equal to that of the majority of its inputs.

Monte Carlo: A class of stochastic (by using pseudorandom numbers) computational algorithms for simulating the behaviour of physical and mathematical systems.

Multiplexing (von Neumann): A scheme for reliable computations based on successive computing stages alternating with random interconnection stages (introduced by von Neumann in 1952).

Redundancy (Factor): Multiplicative increase in the number of (identical) components (subsystems, blocks, gates, devices), which can (automatically) replace (or augment) failing component(s).

Reliability: The ability of a circuit (system, gate, device) to perform and maintain its function(s) under given (as well as hostile or unexpected) conditions, for a certain period of time.

Different Approaches for Cooperation with Metaheuristics

José M. Cadenas
Universidad de Murcia, Spain

Mª Carmen Garrido
Universidad de Murcia, Spain

Enrique Muñoz
Universidad de Murcia, Spain

Carlos Cruz-Corona
Universidad de Granada, Spain

David A. Pelta
Universidad de Granada, Spain

José L. Verdegay
Universidad de Granada, Spain

INTRODUCTION

Working on artificial intelligence, one of the tasks we can carry on is optimization of the possible solutions of a problem. Optimization problems appear. In optimization problems we search for the best solution, or one good enough, to a problem among a lot of alternatives.

Problems we try to solve are usual in daily living. Every person constantly works out optimization problems, e.g. finding the quickest way from home to work taking into account traffic restrictions. Humans can find efficiently solutions to these problems because these are easy enough. Nevertheless, problems can be more complex, for example reducing fuel consumption of a fleet of plains. Computational algorithms are required to tackle this kind of problems. A first approach to solve them is using an exhaustive search. Theoretically, this method always finds the solution, but is not efficient as its execution time grows exponentially.

In order to improve this method heuristics were proposed. Heuristics are intelligent techniques, methods or procedures that use expert knowledge to solve tasks; they try to obtain a high performance referring to solution quality and used resources.

Metaheuristics, term first used by Fred Glover in 1986 (Glover, 1986), arise to improve heuristics, and can be defined as (Melián, Moreno & Moreno, 2003) 'intelligent strategies for designing and improving very general heuristic procedures with a high performance'. Since Glover the field has been extensively developed. The current trend is designing new metaheuristics that improve the solution to given problems. However, another line, very interesting, is reuse existing metaheuristics in a coordinated system. In this article we present two different methods following this line.

BACKGROUND

Several studies have shown that heuristics and metaheuristics are successful tools for providing reasonably good solutions (excellent in some cases) using a moderate number of resources. A brief look at recent literature (Glover & Kochenberger, 2003), (Hart, Krasnogor & Smith, 2004), (Pardalos & Resende, 2002) reveals the wide variety of problems and methods which appear under the overall topic of heuristic optimization. Within this, obtaining strategies which cooperate in a parallel way is an interesting trend. The interest is on account of two reasons: larger problem instances may be solved, and robust tools, that offer high quality solutions despite variations in the characteristics of the instances, may be obtained.

There are different ways of obtaining this cooperation. One way are ant colony systems (Dorigo & Stützle, 2003) and swarm based methods (Eberhart & Kennedy, 2001) appear as one of the first cooperative mechanisms inspired by nature. Nevertheless, the cooperation principle they have presented to date is too rigid for a general purpose model (Crainic & Toulouse, 2003). Another way are parallel metaheuristics, where very interesting perspectives appear. This is the line we will follow.

There have been huge efforts to parallelize different metaheuristics. Thus we may find synchronic implementations of these methods where the information is shared at regular intervals, (Crainic, Toulouse & Gendreau, 1997) using Tabu Search and (Lee & Lee, 1992) using Simulated Annealing. More recently there have been multi-thread asynchronic cooperative implementations (Crainic, Gendreau, Hansen & Mladenovic, 2004) or multilevel cooperative searches (Baños, Gil, Ortega & Montoya, 2004) which, according to the reports in (Crainic & Toulouse, 2003) provide better results than the synchronic implementations.

However, it seems that a cooperative strategy based on a single metaheuristic does not cover all the possibilities and the use of strategies which combine different metaheuristics is recommended. The paper (Le Bouthillier & Crainic, 2005) is a good example. A whole new area of research opens up. Questions such as, 'what will be the role of each metaheuristic?' or 'What cooperation mechanisms should be used?' arise.

Within parallel metaheuristics, we will focus following the classification of (Crainic & Toulouse, 2003) on Multi-search metaheuristics, where several concurrent strategies search the solution space. Among them, we concentrate on those techniques, known as Cooperative multi-search metaheuristics, where each strategy exchanges information with the others during the execution.

Cooperative multi-search metaheuristics obtain better quality solutions than independent methods. But previous studies (Crainic & Toulouse, 2002), (Crainic, Toulouse & Sansó, 2004) demonstrate that cooperative methods with a non-restrictive access to shared information may experiment problems of premature convergence. This seems to be due to the stabilization of the shared information, stabilization caused by the intense exchange of the better solutions. So it would be interesting to find a way of controlling this information exchange.

In this context we propose two approaches in order to control the exchange of information, one using memory to cope with this problem, and the other using a process of knowledge extraction.

The first approach (Pelta, Cruz, Sancho-Royo & Verdegay, 2006) proposes a cooperative strategy where a coordinating agent, modelled by a set of 'ad hoc' fuzzy rules, receives information from a set of solver agents and sends instructions to each of them telling how to continue. Each solver agent implements the Fuzzy Adaptive Neighbourhood Search (FANS) metaheuristic (Blanco, Pelta & Verdegay, 2002) as a clone. FANS is conceived as an adaptive fuzzy neighbourhood based metaheuristic. Its own characteristics allow FANS to capture the qualitative behaviour of several metaheuristics, and thus, can be considered as a "framework" of metaheuristics.

The second approach (Cadenas, Garrido, Liern, Muñoz & Serrano, 2007) uses the same structure but combines a set of different metaheuristics which cooperate within a single coordinated schema, where a coordinating agent modelled by a set of fuzzy rules receives information from the different metaheuristics and sends instructions to each of them. The difference with the previous system lies on the way the rules are obtained. Here, as a result of a knowledge extraction process (Cadenas, Garrido, Hernández & Muñoz, 2006), (Cadenas, Díaz-Valladares, Garrido, Hernández & Serrano, 2006).

TWO COOPERATIVE MULTI-SEARCH METAHEURISTICS

A Cooperative Multi-Search Metaheuristic Using Memory and Fuzzy Rules

The idea of the first strategy can be explained with the help of the diagram in Fig. 1. Given a concrete problem to solve, we have a set of solvers to deal with it. Each solver develops a particular strategy to solve the problem independently, and the whole set of solvers works simultaneously without direct interaction. In order to coordinate the solvers there is a coordinator which knows all the general aspects of the problem concerned and the particular solver features. The coordinator receives reports from the solvers with the obtained results, and returns orders to them.

Figure 1. Diagram of the first strategy
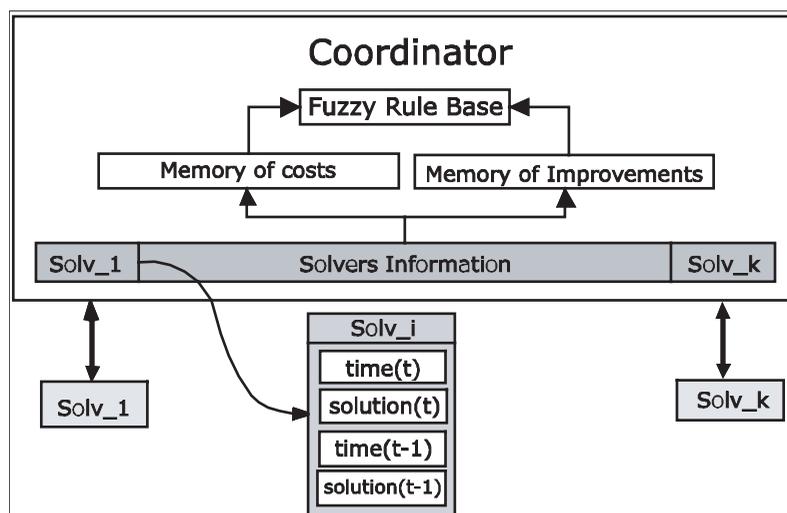
The inner workings of the strategy are quite simple. In the first step, the co-ordinator determines the initial behaviour (set of parameters) for each solver, which models a particular optimization strategy. This behaviour is passed to each solver, which is then executed. For each solver, the co-ordinator keeps the last two reports containing their times, and the corresponding solutions at such times. From the solver information set, the co-ordinator calculates performance measures that will be used to adapt the fuzzy rule base. This fuzzy rule base is generated manually following the principle that *If a solver is working well, keep it; but if a solver seems to be trapped, do something to alter its behaviour*.

Solvers execute asynchronously by sending and receiving information. The co-ordinator checks which solver provided new information and decides whether its behaviour needs to be adapted using the fuzzy rule base. If this is the case, it will obtain a new behaviour and send it to the solvers. Solver operation is quite simple: once execution has begun, performance information is sent and adaptation orders from the coordinator are received alternately.

Each solver thread is implemented by means of the FANS metaheuristic. We use this metaheuristic following three main reasons:

- FANS is essentially a threshold-acceptance local search technique and is therefore easy to understand and implement, and does not require many computational resources.
- FANS can be used as a heuristic template. Each set of parameters implies a different behaviour of the method, and therefore, the qualitative behaviour of other local search techniques can be simulated (Blanco, Pelta & Verdegay, 2002).
- The previous point enables different search schemes to be built, and diversification and intensification procedures to be driven easily. We therefore do not need to implement different algorithms but merely use FANS as a template.

A Cooperative Multi-Search Metaheuristic Using Data Mining to Obtain Fuzzy Rules

The idea of the second strategy is very similar to the first one, and can be seen on Fig. 2. Given a concrete problem to solve, we have a set of solvers to deal with it. Each solver implements a different metaheuristic, and has to solve the problem while coordinates itself with the rest of metaheuristics. In order to perform the cooperation we use a coordinating agent which will con-

D

Figure 2. Diagram of the second strategy
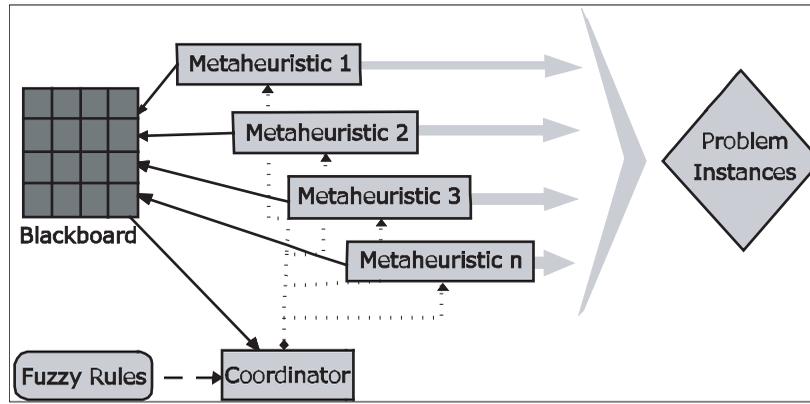
Figure 3. The knowledge extraction process
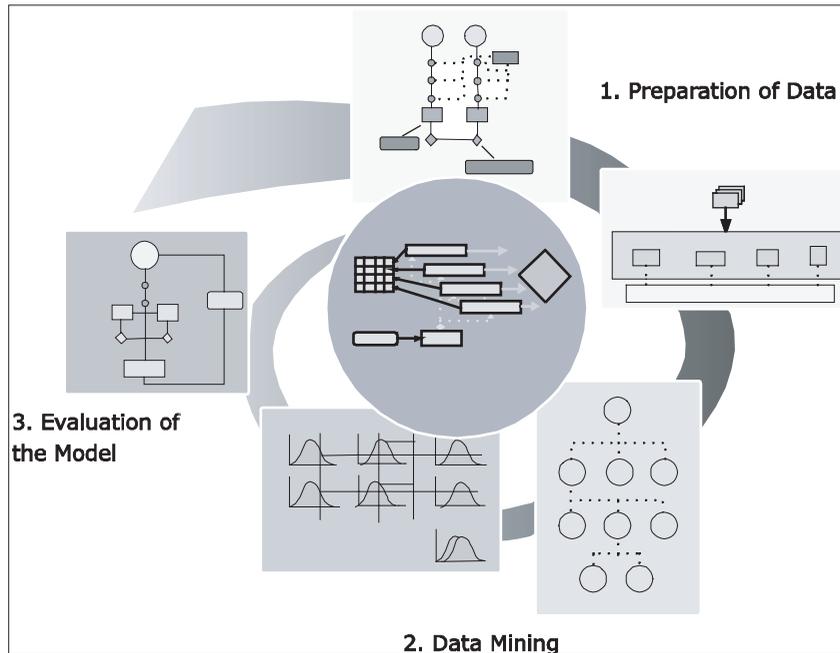
trol and modify the behaviour of the agents. To perform the communication among the different metaheuristics an adapted blackboard model is used. In this model each agent controls a part of the blackboard where writes its performance information, and periodically updates the solution found. The coordinator consults the blackboard in order to monitor the behaviour of

each metaheuristic and to decide how their behaviour has to be modified.

To give intelligence to the coordinator we propose the use of a set of fuzzy rules obtained as a result of the methodological process proposed on (Cadenas, Garrido, Hernández & Muñoz, 2006), (Cadenas, Díaz-Valladares, Garrido, Hernández & Serrano, 2006), and based on a process of knowledge extraction.

The process of knowledge extraction is shown in (Fig. 3) and divided in the following phases:

1. Preparation of data. In this phase we try to obtain a database containing useful information so that Data Mining could be applied. The metaheuristics that are going to be used in the system are chosen and applied to solve sets of instances of the problem, extracting from these executions interesting data such as the parameters of each metaheuristic or the solutions obtained. After that, it is advisable to apply a preprocess to the database in order to obtain those attributes and instances more relevant

2. Data mining. In the second phase, Data Mining techniques are applied to the information obtained in the previous phase in order to get the model of the system coordinator. Firstly a Data Mining technique is chosen, then is applied to the database and finally using the models obtained, a set of fuzzy rules is deduced.

3. Evaluation. In this phase we test the efficiency of the model of the coordinator, and if it is performing efficiently with regard to computational cost and the solutions obtained.

In this paper we present a synchronous implementation of this model where three metaheuristics are used: a genetic algorithm, a tabu search and a simulated annealing.

The system finally obtained, operates as follows: first, the coordinator sets the initial set of parameters for each metaheuristic, according to the knowledge previously extracted. After that each solver starts its search, periodically all the solvers stop and write their solutions, then the coordinator evaluates them and, using the fuzzy rule base, decides how has to change the solutions and parameters of each metaheuristic. To formulate the rule base we used the knowledge extraction process previously defined whose data mining phase was performed using fuzzy decision trees.

Results Obtained by These Approaches

Once we have studied both systems let us show the results obtained by them. Both strategies were tested solving the knapsack problem, whose mathematical formulation is

$$\max \sum_{j=1}^{n} p_j \times x_j$$

$$s.t. \sum_{j=1}^{n} w_j \times x_j \leq C,\ x_j \in \{0,1\}, j=1,\ldots,n$$

where n is the number of items, x_j indicates whether the item j is included in the knapsack or not, p_j is the profit associated with item j, $w_j \in [0,\ldots,r]$ is the weight of the item j, and C is the knapsack capacity. We also assume that $w_j < C$, $\forall j$ (every item fits in the knapsack), and

$$\sum_{j=1}^{n} w_j > C$$

(the whole set of items does not fit).

We chose knapsack problem because of the fact that we can construct test instances varying hardness according to three characteristics: instance size, type of correlation between weights and profits, and range of the values available for the weights.

We finally carried out the tests with an implementation of each system. The implementation of the memory based system used six solvers, each one implementing FANS with different parameters, and being executed for 30 seconds (Memory based in table 1). The implementation of the data mining based system, as said before, used three solvers, each one implementing a different metaheuristic (a genetic algorithm, a tabu search and a simulated annealing), and being executed for 60 seconds (DM based in table 1). In order to test the performance of the systems we also executed each metaheuristic individually (FANS, Tabu Search, Simulated Annealing, Genetic Algorithm) for 180 seconds.

In table 1 we show the average error obtained for different types and sizes of instances, comparing the Memory based approach with individual FANS, and the DM based with the average results of the three metaheuristics that compose it. As we can see each strategy outperforms its components.

FUTURE TRENDS

After this work several lines of research arise. Related to first strategy, the topic of what kind of information

D

Table 1. Average error for knapsack problem

		Memory based	FANS	DM based	Avg. Individual
NC	1000	2.32	8.7	0,84	3,2
	2000	2.04	14.15	1,32	4,31
SC	1000	0.94	3.16	0,93	3,6
	2000	1.69	3.81	2,29	5,38
Circle	1000	6.73	10.42	5,14	15,44
	2000	8.10	13.41	12,11	18,26

is stored in the coordinator memory and how this is used to control the global search behavior of the strategy. Related to the second, the knowledge extraction process needs to be improved and tested with different data mining techniques. And to both strategies, the improvement of the fuzzy rule base is another topic to be addressed.

One last consideration is the application field. The knapsack problem is considered one of the "easiest" NP-hard problems, so it would be interesting to apply these strategies to more complex problems such as the p-median, p-hub median or the protein structure comparison.

CONCLUSION

This paper proposes two strategies to cope with convergence problems showed by cooperative multi-search metaheuristics. The first strategy suggests the use of memory in order to define a set of fuzzy rules which control the exchanges of solutions associated to a coordinated schema where similar metaheuristics cooperate. The second strategy proposes the use of a knowledge extraction process to obtain a set of fuzzy rules which control the exchanges of solutions of a coordinated system where different metaheuristics cooperate. Both approaches have been tested and have shown their good performance.

ACKNOWLEDGMENT

REFERENCES

Baños, R., &. Gil, C, & Ortega, J., & Montoya, F.G. (2004). *A Parallel Multilevel Metaheuristic for Graph Partitioning*. Journal of Heuristics, 10(3), 315–336.

Blanco, A., & Pelta, D., & Verdegay, J.L. (2002). *A fuzzy valuation-based local search framework for combinatorial problems*. Fuzzy Optimization and Decision Making, 1(2), 177–193.

Cadenas, J.M., & Garrido, M.C., & Hernández, L.D., & Muñoz, E. (2006). *Towards the definition of a Data Mining process based in Fuzzy Sets for Cooperative Metaheuristic systems*. In Proceedings from IPMU2006: The 11th Information Processing and Management of Uncertainty in Knowledge-Based systems International Conference (pp. 2828–2835). Paris, France.

Cadenas, J.M., & Díaz-Valladares, R.A., & Garrido, M.C., & Hernández, L.D., & Serrano E. (2006). *Modelado del Coordinador de un sistema Meta-Heurístico Cooperativo mediante SoftComputing*. In Proceedings from CMPI2006: Campus Multidisciplinar en Percepción e Inteligencia (pp. 679–689). Albacete, Spain.

Cadenas, J.M., & Garrido, M.C., & Liern, V. & Muñoz, E., & Serrano E. (2007). *Un prototipo del coordinador de un Sistema Metaheurístico Cooperativo para el Problema de la Mochila.* In Proceedings from MAEB'07: V congreso español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (pp. 811–818). Tenerife, Spain.

Crainic, T.G., & Toulouse, M., & Gendreau, M. (1997). *Towards a taxonomy of parallel tabu search algorithms.* INFORMS Journal on Computing, 9(1), 61–72.

Crainic, T.G., & Toulouse, M. (2002). *Cooperative parallel tabu search for capacitated network design.* Journal of Heuristics, 8(6), 601–627.

Crainic, T.G., & Toulouse, M. (2003). *Parallel Strategies for Metaheuristics.* In F. Glover & G.A. Kochenberger (Eds.), Handbook of Metaheuristics (pp. 475–514). London: Kluwer Academic Publisher.

Crainic, T.G., & Toulouse, M., & Sansó, B. (2004). *Systemic behavior of cooperative search algorithms.* Parallel Computing, 30(1), 57–79.

Crainic, T.G., & Gendreau, M., & Hansen, P., & Mladenovic, N. (2004). *Cooperative parallel variable neighborhood search for the p-median.* Journal of Heuristics, 10(3), 293–314.

Dorigo, M., & Stützle, T. (2003). *Ant Colony Optimization Metaheuristic.* In F. Glover & G.A. Kochenberger (Eds.), Handbook of Metaheuristics (pp. 251–286). London: Kluwer Academic Publisher.

Eberhart, R., & Kennedy, J. (Eds.) (2001). *Swarm Intelligence.* London: Academic Press.

Glover, F. (1986). *Future paths for integer programming and links to artificial intelligence.* Computers and Operations Research, 13(5), 533–549.

Glover, F., & Kochenberger G.A. (Eds.) (2003). *Handbook of Metaheuristics.* London: Kluwer Academic Publishers.

Hart, W., & Krasnogor, N., & Smith, J. (Eds.) (2004). *Recent Advances in Memetic Algorithms.* Studies in Fuzziness and Soft Computing, Physica-Verlag, Wurzburg.

Le Bouthillier, A., & Crainic, T.G. (2005). *A cooperative parallel meta-heuristic for the vehicle routing problem with time windows.* Computers and Operations Research, 32(7), 1685–1708.

Lee, K., & Lee, S. (1992). *Efficient parallelization of simulated annealing using multiple markov chains: an application to graph partitioning.* In Proceedings from ICPP'02: International Conference on Parallel Processing (pp. 177–180). Vancouver, Canada.

Melián, B., & Moreno, J.A., & Moreno, J.M. (2003). *Metaheurísticas: una visión global.* Revista Iberoamericana de Inteligencia Artificial, 19, 7–28.

Pardalos, P., & Resende, M. (Eds.) (2002). *Handbook of Applied Optimization.* Oxford: Oxford University Press.

Pelta, D., & Cruz, C., & Sancho-Royo, A., & Verdegay, J.L. (2006). *Using memory and fuzzy rules in a cooperative multi-thread strategy for optimization.* Information Sciences, 176(13), 1849–1868.

KEY TERMS

Blackboard: A shared repository of problems, partial solutions, suggestions, and contributed information. The blackboard can be seen as a dynamic "library" of contributions to the current problem that have been recently "published" by other knowledge sources.

Cooperative Multi-Search Metaheuristics: A parallelization strategy for metaheuristics in which parallelism is obtained from multiple concurrent explorations of the solution space and where metaheuristics exchange information during their execution in order to cooperate.

Data Mining: The most characteristic stage of the Knowledge Extraction process, where the aim is to produce new useful knowledge by constructing a model from the data gathered for this purpose.

Fuzzy Rules: Linguistic if-then constructions that have the general form "if A then B" where A and B are collections of propositions containing linguistic variables (A is called the premise and B is the consequence). The use of linguistic variables and fuzzy if-then rules exploits the tolerance for imprecision and uncertainty.

Heuristic: A method or approach that tries to apply expert knowledge in the resolution of a problem with the aim of increasing the probability of solving it.

Knowledge Extraction/Discovery: The non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns from large data collections. The overall process and discipline of extracting useful knowledge and includes data warehousing, data cleansing and data manipulation tasks right through to the interpretation and exploitation of results.

Metaheuristic: A high-level strategy for solving a very general class of computational problems by combining user given black-box procedures — usually heuristics — in a hopefully efficient way.

Optimization Problem: A computational problem whose object is to find the best from all feasible solutions.

Parallel Metaheuristics: Metaheuristics in which different threads search concurrently the solution space. They appear naturally in develop of metaheuristics as a way of improving the acceleration factor in the search of solutions.

Problem Instance: A concrete representation of a problem with characteristics that distinguish it from the rest.

D

Differential Evolution with Self–Adaptation

Janez Brest
University of Maribor, Slovenia

INTRODUCTION

Many practical engineering applications can be formulated as a global optimization problem, in which objective function has many local minima, and derivatives of the objective function are unavailable. Differential Evolution (DE) is a floating-point encoding evolutionary algorithm for global optimization over continuous spaces (Storn & Price, 1997) (Liu & Lampinen, 2005) (Price, Storn & Lampinen, 2005) (Feoktistov, 2006). Nowadays it is used as a powerful global optimization method within a wide range of research areas.

Recent researches indicate that self-adaptive DE algorithms are considerably better than the original DE algorithm. The necessity of changing control parameters during the optimization process is also confirmed based on the experiments in (Brest, Greiner, Bošković, Mernik, Žumer, 2006a). DE with self-adaptive control parameters has already been presented in (Brest et al., 2006a).

This chapter presents self-adaptive approaches that were recently proposed for control parameters in DE algorithm.

BACKGROUND

Differential Evolution

DE creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better fitness value.

The population of the original DE algorithm (Storn & Price, 1995) (Storn & Price, 1997) contains *NP D*-dimensional vectors: $\mathbf{x}_{i,G}$, $i = 1, 2, …, NP$. *G* denotes the generation. The initial population is usually selected uniform randomly between the lower and upper bounds. The bounds are specified by the user according to the nature of the problem. After initialization DE performs several vector transforms (operations): mutation, crossover, and selection.

Mutant vector $\mathbf{v}_{i,G}$ can be created by using one of the mutation strategies (Price et al., 2005). The most useful strategy is 'rand/1': $\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + \mathrm{F} \times (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G})$, where *F* is the mutation scale factor within range [0, 2], usually less than 1. Indexes *r1*, *r2*, *r3* represent the random and distinct integers generated within range [1, *NP*], and also different from index *i*.

After mutation, a 'binary' crossover operation forms the trial vector $\mathbf{u}_{i,G}$, according to the ith population vector and its corresponding mutant vector $\mathbf{v}_{i,G}$:

$$\underline{\text{if }} (rand \le CR \text{ } \underline{\text{or }} j = j_{rand}) \text{ } \underline{\text{then }} u_{i,j,G} = v_{i,j,G} \text{ } \underline{\text{else }} u_{i,j,G} = x_{i,j,G} \text{ ,}$$

where $i = 1, 2, …, NP$ and $j = 1, 2, …, D$. *CR* is the crossover parameter or factor within the range [0,1] and presents the probability of creating parameters for the trial vector from the mutant vector. Uniform random value *rand* is within [0, 1]. Index $j_{rand} \in [1, NP]$ is a randomly chosen index and is responsible for the trial vector containing at least one parameter from the mutant vector.

The selection operation selects, according to the objective fitness value of the population vector $\mathbf{x}_{i,G}$ and its corresponding trial vector $\mathbf{u}_{i,G}$, which vector will survive to be a member of the next generation.

The original DE has more strategies and Feoktistov (Feoktistov, 2006) proposed some general extensions to DE strategies. The question is which strategy is the most suitable to solve a particular problem. Recently some researchers used various combinations of two, three or even more strategies during the evolutionary process.

Parameter Tuning and Parameter Control

Globally, we distinguish between two major forms of setting parameter values: parameter *tuning* and parameter *control* (Eiben, Hinterding & Michalewicz, 1999). The former means the commonly practiced approach that tries to find good values for the parameters before running the algorithm, then tuning the algorithm using

D

these values, which remain fixed during the run. The latter means that values for the parameters are changed during the run. According to Eiben *et al.* (Eiben et al., 1999) (Eiben & Smith, 2003), the change can be categorized into three classes:

1.   *Deterministic parameter control* takes place when the value of a parameter is altered by some deterministic rule.
2.   *Adaptive parameter control* is used when there is some form of feedback from the search that is used to determine the direction and/or the magnitude of the change to the parameter.
3.   *Self-adaptive parameter control* is the idea that "evolution of the evolution" can be used to implement the self-adaptation of parameters. Here, the parameters to be adapted are encoded into the chromosome (individuals) and undergo the actions of genetic operators. The better values of these encoded parameters lead to better individuals which, in turn, are more likely to survive and produce offspring and, hence, propagate these better parameter values.

DE has three control parameters: amplification factor of the difference vector - *F*, crossover control parameter - *CR*, and population size - *NP*. The original DE algorithm keeps all three control parameters fixed during the optimization process. However, there still exists a lack of knowledge about how to find reasonably good values for the control parameters of DE, for a given function (Liu & Lampinen, 2005).

Although the DE algorithm has been shown to be a simple, yet powerful, evolutionary algorithm for optimizing continuous functions, users are still faced with the problem of preliminary testing and hand-tuning of its control parameters prior to commencing the actual optimization process (Teo, 2006). As a solution, self-adaptation has proved to be highly beneficial for automatically and dynamically adjusting control parameters. Self-adaptation allows an evolutionary strategy to adapt itself to any general class of problem, by reconfiguring itself accordingly, and does this without any user interaction (Bäck, 2002) (Bäck, Fogel & Michalewicz, 1997) (Eiben, Hinterding & Michalewicz, 2003).

RELATED WORK

Work Releted to Differential Evolution

The DE (Storn & Price, 1995) (Storn & Price, 1997) algorithm was proposed by Storn and Price, and since then it has been used in many practical cases. The original DE was modified and many new versions have been proposed. Ali and Törn (Ali & Törn, 2004) proposed new versions of the DE algorithm, and also suggested some modifications to the classical DE, in order to improve its efficiency and robustness. They introduced an auxiliary population of *NP* individuals alongside the original population (noted in (Ali, 2004), a notation using sets is used). Next they proposed a rule for calculating the control parameter *F*, automatically. Jiao et al. (Jiao, Dang, Leung & Hao, 2006) proposed a modification of the DE algorithm, applying a number-theoretical method for generating the initial population, and using simplified quadratic approximation with the three best points. Mezura-Montes et al. (Mezura-Montes, Velázquez-Reyes & Coello Coello, 2006) conducted a comparative study of DE variants. They proposed a rule for changing control parameter *F* at random from interval [0.4, 1.0] at generation level. They used different values of control parameter *CR* for each problem. The best *CR* value for each problem was obtained by additional experimentation. Tvrdik in (Tvrdik, 2006) proposed a DE algorithm using competition between different control parameter settings. The prominence of the DE algorithm and its applications is shown in recently published books (Price et al., 2005), (Feoktistov, 2006). Feoktistov in his book (Feoktistov, 2006, p. 18) says, that "the concept of differential evolution is a spontaneous self-adaptability to the function".

Work Releted to Adaptive or Self-Adaptive DE

Liu and Lampinen (Liu & Lampinen, 2005) proposed a version of DE, where the mutation control parameter and the crossover control parameter are adaptive. A self-adaptive DE (SDE) is proposed by Omran et al. (Omran, Salman & Engelbrecht, 2005) (Salman, Engelbrecht & Omran, 2007), where parameter tuning is not required. Self-adapting was applied for control

parameters *F* and *CR*. Teo (Teo, 2006) made an attempt at self-adapting the population size parameter, in addition to self-adapting crossover and mutation rates. Brest et al. (Brest et al., 2006a) proposed a DE algorithm, using a self-adapting mechanism on the control parameters *F* and *CR*. The performance of the self-adaptive differential evolution algorithm was evaluated using a set of benchmark functions provided for constrained real parameter optimization (Brest, Žumer & Sepesy Maučec, 2006b). Qin and Suganthan (Qin & Suganthan, 2005) proposed the "Self-adaptive Differential Evolution algorithm (SaDE), where the choice of learning strategy and the two control parameters *F* and *CR* do not require pre-defining. During evolution, suitable learning strategy and parameter settings are gradually self-adapted, according to the learning experience." Brest et al. (Brest, Bošković, Greiner, Žumer & Sepesy Maučec, 2007) reported performance comparison of certain selected DE algorithms, which use different self-adaptive or adaptive control parameter mechanisms. In this paper the DE algorithms used more than one of the DE strategies (Price et al., 2005), (Feoktistov, 2006). Self-adaptation has been used extensively in evolutionary programming (Fogel, 1995) and evolution strategies (ES) (Bäck & Schwefel, 1993) to adjust the search step size for each objective variable (Liang, Yao & Newton, 2001). Abbass (Abbass, 2002) proposed self-adaptive DE for multi-objective optimization problems.

SELF-ADAPTIVE CONTROL PARAMETERS IN DIFFERENTIAL EVOLUTION

This section presents three Self-Adaptive DE approaches, which has been applied to the control parameters *F* and *CR*.

The Self-Adaptive Control Parameters Using Uniform Distribution

The Self-Adaptive DE refers to the self-adapting mechanism on the control parameters, proposed by Brest et al. (Brest et al., 2006a). This self-adapting mechanism used *'rand/1/bin'* strategy. Each individual in the population was extended using the values of two control parameters: $(\mathbf{x}_{i,G}, F_{i,G}, CR_{i,G})$, $i \in 1, 2, ...,$

NP. Both of the control parameters were applied at individual level.

Brest et al. (Brest et al., 2006a) proposed a self-adaptive DE where new control parameters $F_{i,G+1}$ and $CR_{i,G+1}$ are calculated as follows:

$$\underline{\text{if }} rand_1 < \tau_1 \underline{\text{ then }} F_{i,G+1} = F_l + rand_2 \times F_u \underline{\text{ else }} F_{i,G+1} = F_{i,G},$$

$$\underline{\text{if }} rand_3 < \tau_2 \underline{\text{ then }} CR_{i,G+1} = rand_4 \underline{\text{ else }} CR_{i,G+1} = CR_{i,G},$$

and they produce control parameters *F* and *CR* in a new vector. The quantities $rand_j$, $j \in \{1, 2, 3, 4\}$ are uniform random values $\in [0, 1]$. The quantities τ_1 and τ_2 represent the probabilities of adjusting control parameters *F* and *CR*, respectively. The parameters τ_1, τ_2, F_l, F_u were taken as fixed values 0.1, 0.1, 0.1, 0.9, respectively. The new *F* takes a value from [0.1,1.0] in a random manner. The new *CR* takes a value from [0,1]. The new $F_{i,G+1}$ and $CR_{i,G+1}$ are obtained before the mutation is performed. So they influence the mutation, crossover and selection operations of the new vector $\mathbf{x}_{i,G+1}$.

In (Brest et al., 2006a) a self-adaptive control mechanism was used to change the control parameters *F* and *CR* during the evolutionary process. The third control parameter *NP* was kept unchanged.

Abbass's Approach

Abbass (Abbass, 2002) proposed Self-adaptive Pareto Differential Evolution (SPDE) algorithm. The SPDE was used for multi-objective optimization problems. New control parameters $F_{i,G+1}$ and $CR_{i,G+1}$ are calculated as follows:

$$F_{i,G+1} = N(0,1),$$

$$CR_{i,G+1} = CR_{r1,G} + N(0,1) \times (CR_{r2,G} - CR_{r3,G}),$$

where $N(0,1)$ is Gaussian distribution. If $F_{i,G+1}$ value is not in [0,1] then simple rule is used to repair it. And similar for $CR_{i,G+1}$ value. Then mutant vector $\mathbf{v}_{i,G}$ is calculated:

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F_{i,G+1} \times (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G})$$

and crossover operation is performed:

$$\underline{\text{if}} \ (rand \leq CR_{i,G+1} \ \underline{\text{or}} \ j = j_{rand}) \ \ \underline{\text{then}} \ \ u_{i,j,G} = v_{i,j,G} \ \ \underline{\text{else}}$$
$$u_{i,j,G} = x_{i,j,G} \, ,$$

The control parameter CR is self-adapted, by encoding it into each individual.

Approach Proposed by Omran, Salman and Engelbrecht

Due to the success achieved in SPDE by self-adapting CR, Omran et al. (Omran, Salman & Engelbrecht, 2005) (Salman, Engelbrecht & Omran, 2007) proposed a self-adaptive DE (SDE), where the same mechanism is applied to self-adapt the control parameter F. The control parameter CR is generated for each individual from a normal distribution ($CR \sim N(0.5, 0.15)$) in SDE and the mutation operation changes as follows:

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F_{i,G+1} \times (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}),$$

where

$$F_{i,G+1} = F_{r4,G} + N(0,0.5) \times (F_{r5,G} - F_{r6,G}).$$

Indexes $r4, r5, r6$ represent the random and distinct integers generated within range [1, NP]. Thus, each individual i has its own control parameter F_i which is calculated as a stochastic linear combination of the control parameters of randomly selected individuals.

The presented self-adapting mechanisms on the control parameters F and CR, use *'rand/1/bin'* strategy. Both of the control parameters are applied at individual level. The third control parameter NP remains fixed during the evolutionary process.

FUTURE TRENDS

The behaviour of DE is influenced by values of its parameters (F, CR, NP). During last two decades a lot of papers addressed the problem of finding insight concerning the behaviour of the algorithm (Zaharie, 2002). The theory of DE is still behind the empirical studies. The theoretical studies of DE are highly desirable as future researches.

It is not an easy task for one optimization algorithm to be both fast (e.g. it needs a small number of function evaluations) and robust (e.g. it does not get trapped in local optimum) at the same time. Based on our experiences (other authors reported similar observations) with the DE algorithm, we can conclude that DE provides more robustness if the population size is higher. If the population size is increased, on the other hand, more computational power (on average) is needed. The population size is an important control parameter, and thus adaptive and/or self-adaptive approaches on this parameter are expected in the future.

In this chapter only the *'rand/1/bin'* DE strategy is used, but the DE algorithm has more strategies (Price et al., 2005), (Feoktistov, 2006). Which combination of (self-adaptive) DE strategies should someone use to get the best performances. One can use the involved strategy with the same probability, or with different probability, or even use a self-adaptation to choose the most suitable strategy during the optimization process.

Future work may also be directed towards testing the proposed self-adaptive versions of the DE algorithm, especially on constrained optimization problems. Multi-objective optimization is also a challenge for future work.

CONCLUSION

This chapter carried out differential evolution (DE) algorithm with focus on the self-adaptive control parameters. Three self-adaptive approaches, which were recently proposed in literature, are described in the chapter. The presented approaches have control parameters applied at individual level. If we look in literature, the self-adaptive versions of the DE algorithm usually gave better performance results in comparison to the original DE algorithm. We can conclude that self-adaptation can improve the performance of the DE algorithm and this powerful global optimization algorithm could be used over a wide-range of research areas in the future.

REFERENCES

Abbass, H. (2002). The Self-Adaptive Pareto Differential Evolution Algorithm. *Proceedings of the 2002 Congress on Evolutionary Computation*. 831-836.

Ali, M.M. & Törn, A. (2004). Population Set-Based Global Optimization Algorithms: Some Modifications

and Numerical Studies. *Computers & Operations Research.* 31(10), 1703-1725.

Bäck, T. (2002) Adaptive Business Intelligence Based on Evolution Strategies: Some Application Examples of Self-Adaptive Software. *Information Sciences.* 148(1-4), 113-121.

Bäck, T., Fogel, D.B., & Michalewicz, Z. (Editors) (1997). *Handbook of Evolutionary Computation.* Institute of Physics Publishing and Oxford University Press.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, New York.

Bäck, T., & Schwefel, H.-P. (1993) An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation.* 1(1), 1-23.

Brest, J., Bošković, B., Greiner, S., Žumer, V., & Sepesy Maučec, M. (2007). Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft Computing - A Fusion of Foundations, Methodologies and Applications. 11(7), 617-629.* DOI 10.1007/s00500-006-0124-0.

Brest, J., Greiner, S., Bošković, B., Mernik, M., & Žumer, V. (2006). Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation.* 10(6), 646-657. DOI 10.1109/TEVC.2006.87213.

Brest, J., Žumer, V., & Sepesy Maučec, M. (2006). Self-adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization. *The 2006 IEEE Congress on Evolutionary Computation, CEC2006.* 919-926. IEEE Press.

Eiben, A.E., Hinterding, R., & Michalewicz, Z. (1999). Parameter Control in Evolutionary Algorithms, *IEEE Transactions on Evolutionary Computation.* 3(2), 124-141.

Eiben, A.E., & Smith, J.E. (2003). Introduction to Evolutionary Computing. *Natural Computing.* Springer-Verlag, Berlin.

Fan, H.-Y., & Lampinen, J. (2003). A Trigonometric Mutation Operation to Differential Evolution. *Journal of Global Optimization.* 27(1), 105-129.

Feoktistov, V. (2006). *Differential Evolution: In Search of Solutions.* Springer, New York.

Fogel, D. (1995). A Comparison of Evolutionary Programming and Genetic Algorithms on Selected Constrained Optimization Problems. *Simulation,* 64(6), 397-404.

Jiao, Y.-C., Dang, C., Leung, Y., & Hao, Y. (2006). A modification to the new version of the price's algorithm for continuous global optimization problems. *Journal of Global Optimization.* 36(4), 609-626. DOI 10.1007/s10898-006-9030-3.

Lee, C.Y., & Yao, X. (2004). Evolutionary Programming Using Mutations Based on the Levy Probability Distribution, *IEEE Transactions on Evolutionary Computation.* 8(1), 1-13.

Liang, K.-H., Yao, X., & Newton, C.S. (2001). Adapting Self-adaptive Parameters in Evolutionary Algorithms. *Applied Intelligence.* 15(3), 171-180.

Liu, J., & Lampinen, J. (2005). A Fuzzy Adaptive Differential Evolution Algorithm. *Soft Computing - A Fusion of Foundations, Methodologies and Applications.* 9(6), 448-462.

Mezura-Montes, E., Velázquez-Reyes, J., & Coello Coello, C.A. (2006). A comparative study of differential evolution variants for global optimization. *Genetic and Evolutionary Computation Conference GECCO2006.* 485-492.

Ohkura, K., Matsumura, Y., & Ueda, K. (2001). Robust Evolution Strategies. *Applied Intelligence.* 15(3), 153-169.

Omran, M.G.H., Salman, A., Engelbrecht, A.P. (2005). Self-adaptive Differential Evolution, *Lecture Notes in Computer Science.* 3801, 192-199.

Price, K.V., Storn, R.M., & Lampinen, J.A. (2005). *Differential Evolution, A Practical Approach to Global Optimization.* Springer.

Qin, A.K., & Suganthan, P.N. (2005). Self-adaptive Differential Evolution Algorithm for Numerical Optimization, *The 2005 IEEE Congress on Evolutionary Computation CEC2005.* 1785-1791. IEEE Press. DOI: 10.1109/CEC.2005.1554904.

Rönkkönen, J., Kukkonen, S., & Price, K.V. (2005). Real-Parameter Optimization with Differential Evolu-

tion, *The 2005 IEEE Congress on Evolutionary Computation CEC2005*. 506-513. IEEE Press.

Salman, A., Engelbrecht, A.P., Omran, M.G.H., (2007). Empirical analysis of self-adaptive differential evolution, *European Journal of Operational Research*. 183, 785-804.

Storn, R., & Price, K. (1995). Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Berkeley, CA.

Storn, R., & Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*. 11, 341-359.

Teo, J. (2006). Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*. 10(8), 673-686. DOI: 10.1007/s00500-005-0537-1.

Tvrdik, J. (2006). Competitive differential evolution, *MENDEL'06, 12th International Conference on Soft Computing*. 7-12.

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary Programming Made Faster, *IEEE Transactions on Evolutionary Computation*. 3(2), 82-102.

Zaharie, D. (2002). Critical values for the control parameters of differential evolution algorithms, *MENDEL'02, 8th International Conference on Soft Computing*. 62-67.

KEY TERMS

D

Area of the Search Space: Set of specific ranges or values of the input variables that constitute a subset of the search space.

Control Parameter: Control parameter determines behaviour of evolutionary program (e.g. population size).

Differential Evolution: An evolutionary algorithm for global optimization, which realized the evolution of a population of individuals in a manner that uses of differences between individuals.

Evolutionary Computation: Solution approach guided by biological evolution, which begins with potential solution models, then iteratively applies algorithms to find the fittest models from the set to serve as inputs to the next iteration, ultimately leading to a model that best represents the data.

Individual: An individual represents a candidate solution. During the optimization process an evolutionary algorithm usually uses a population of individuals to solve a particular problem.

Search Space: Set of all possible situations of the optimization problem that we want to solve.

Self-Adaptation: The ability that allows an evolutionary algorithm to adapt itself to any general class of problems, by reconfiguring itself accordingly, and do this without and user interaction.

Discovering Mappings Between Ontologies

Vikram Sorathia
Dhirubhai Ambani Institute of Information and Communication Technology, India

Anutosh Maitra
Dhirubhai Ambani Institute of Information and Communication Technology, India

INTRODUCTION

Knowledge Representation is important part of AI. The purpose is to reveal best possible representation of the Universe of Discourse (UoD) by capturing entities, concepts and relations among them. With increased understanding of various scientific and technological disciplines, it is possible to derive rules that governs the behaviour and outcome of the entities in the UoD. In certain cases, it is not possible to establish any explicit rule, yet through experience or observation, some experts can define rules from their tacit knowledge in specific domain.

Knowledge representation techniques are focused on techniques that allows externalization of implicit and explicit knowledge of expert(s) with a goal of reuse in absence of physical presence of such expertise. To ease this task, two parallel dimensions have developed over period of time. One dimension is focused on investigating more efficient methods that best suit the knowledge representation requirement resulting in theories and tools that allows capturing the domain knowledge (Brachman & Levesque, 2004). Another development has taken place in harmonization of tools and techniques that allows standard based representation of knowledge (Davies, Studer, & Warren, 2006).

Various languages are proposed for representation of the knowledge. Reasoning and classification algorithms are also realized. As an outcome of standardization process, standards like DAML-OIL (Horrocks & Patel-Schneider, 2001), RDF (Manola & Miller, 2004) and OWL(Antoniou & Harmelen, 2004) are introduced. Capturing the benefit of both developments, the tooling is also came in to existence that allows creation of knowledgebase.

As a result of these developments, the amount of publicly shared knowledge is continuously increasing. At the time of this writing, a search engine like Swoogle (Ding et al., 2004)-developed to index publicly available Ontologies, is handling over 2,173,724 semantic web documents containing 431,467,096 triples.

While the developments are yielding positive results by such a huge amount of knowledge available for reuse, it have become difficult to select and reuse required knowledge from this vast pool. The concepts and their relations that are important to the given problem could have already been defined in multiple Ontologies with different perspectives with specific level of details. It is very likely that to get complete representation of the knowledge, multiple Ontologies must be utilized. This requirement has introduced a new discipline within the domain of knowledge representation that is focused on investigation of techniques and tools that allows integration of multiple shared Ontologies.

BACKGROUND

The problem of Ontology integration is not completely new. Schema Matching is a similar problem being addressed in the context of enterprise integration. But, in Ontology matching, the scale and complexity is much higher and requires special considerations. (Shvaiko & Euzenat, 2006) highlights the key similarities and differences between both the techniques. In schema matching, the semantics of the given term is guessed whereas the ontology matching methods relies on deriving the semantics from explicit representation of concepts and relations in given Ontology. Numerous methods and approaches have been proposed that attempt to solve the problem targeting specific aspects of the represented knowledge(Ehring, 2007).

Apart from standards that guide the languages used for the development of Ontology, some standard Ontologies have also been defined. The role of these Ontologies is to provide framework of vary basic elements and their relations, based on which complex domain knowledge can be developed. SUO(Niles & Pease,

2001), SUMA(Niles & Pease, 2003), OpenCyc(Sicilia et al., 2004) are examples of the same. SWEET (Raskin, 2003) provides standard Ontologies in environmental science domain. Hence, the levels in Ontology also address important dimension in knowledge engineering through integrating available Ontologies.

ONTOLOGY MAPPING TECHNIQUES

Research in integration of multiple Ontologies have resulted in various techniques and tools that have successfully demonstrated capabilities in producing the required results(Noy, 2004b). The ontology integration is addressed as Ontology mapping, matching, merging, transforming and other such activities. The integration is achieved by focusing on finding similarities among the concepts of separate Ontologies. The similarity or nearness can be established by employing various techniques, and numerous such approaches have been published demonstrating the suitability of single or hybrid approaches. The taxonomic overview of existing methodology is provided in many survey papers that provides a reasonable entry in to the domain of Ontology integration. (Kalfoglou et al., 2005) provides comprehensive survey of Ontology mapping approach and classify them on Semantic Intensity Spectrum. (Noy, 2004a) (Kalfoglou & Schorlemmer, 2005)and (Predoiu et al., 2006) provides comprehensive survey discussing state-of-the-art of present research efforts.

Ontologies consists of concepts and elements. The integration process that establishes the similarity among concepts consists of three dimensions (Shvaiko & Euzenat, 2006). The input dimension is related to underlying data model and can operate at schema level or instance level. Second is the process dimension that classifies approach as exact or approximate determination. Third dimension deals with output in the form of Cardinality, type of relation and the confidence. Integration can be done by identification of Alignment.

Concept Level Approaches

Concept level approaches are restricted only to the name of the concept and employ various methods to match whole or part of the concept names that belong to different Ontologies. Though these syntax oriented approaches proves to be less efficient when applied in isolation, they are generally employed in pre-integra-

tion preparation phase (or normalization phase) of more complex semantic oriented approaches. Many of the Schema Matching techniques are directly applicable for concept level approaches.

String Level Concept Matching

It is based on the simple assumption that concept having similarity is represented with same name in different Ontologies. Upon identification of such string level similarity the source Ontologies can either mapped or merged. PROMPT(Noy & Musen, 2000) Ontology Merging tool employs string level concept matching approach.

Sub-String Level Concept Matching

Approaches that brakes the input concepts in to smaller segments on the basis of prefix, suffix and other structures. Another approach establishes the similarity by identifying the *Edit Distance*. For example if Nikon and NKN are under consideration, the Edit Distance is a number of insertion, deletion and substitution of characters that will be required in Nikon and NKN to transform one into the other. N-gram technique is employed for deriving a set of substrings by selecting n number of characters from input string. For example trigram of NIKON results in NIK, IKO and KON. The derived set can further be subjected to simple string matcher for finding similarities.

Lexical Matching

Lexical approaches are employed to identify and extract tokens from the input string. This is particularly useful when concept name are created using mix of alpha-numeric characters that can be processed to separate operators, numbers, punctuations and other types of token to reveal processable substrings. LOM(Li, 2004)- a Lexicon based Ontology Mapping tool employs strategy to determine similarity by matching the whole term, word constituent, synset, and type matching (Choi et al., 2006). OLA(Euzenat et al., 2005) and Cupid(Madhavan et al., 2001) also employs lexical techniques for finding similarity among concepts.

Linguistic Similarity Approach

Natural Language Processing domain offers various text processing techniques such as stemming, tokenization, tagging, elimination, expansion etc. that can improve result of similarity finding effort. Usage and grammar of language may result in mismatch for example, like Product and Products, in such cases Lemmatization can be used. Tokenization that can remove grammatical elements from concept name can be utilized. Unnecessary articles, Preposition, conjunction and other features can be removed using Elimination technique. Lexical relation can be also identified using sense based approach by exploring Hypernyms, hyponyms etc. in WordNet(Miller, 1995). HCONE-Merge(Kotis et al., 2006) is a Ontology merging approach that uses Latent Semantic Analysis (LSA) technique and carryout lookup WordNet and by expanding word sense by hyponymy. Domain or Application specific terminology can also be integrated for disambiguation. Cupid - a schema matching approach-employs linguistic matching in its initial phase. Quick Ontology Mapping (Ehrig & Sure, 2005) employs finding of linguistic similarity of concepts. ASCO(Le et al., 2004) technique calculate linguistic similarity as a linear combination of name, label and description similarity of concepts.

Semantic Concept Matching Approach

iMAP(Dhamankar et al., 2004) employs semantic matching for integration of heterogeneous data sources. It addresses 1-1 and complex matches among concepts. The multiple search approach toward identification of matches, utilizes domain knowledge for improving schema matching accuracy. S-Match (Giunchiglia et al., 2005) derives semantic matching between two graph structures. MAFRA (Maedche et al., 2002) (MApping FRAmework) employs Semantic Bridge and Service Centric approach.

Pairs

By transforming the input Ontologies into directed labelled graph, the Similarity Flooding techniques generates Pair-wise Connectivity Graphs (PCG) where a node consists of a pair or matching elements from the sources. The technique further assigns weights to the edges indicating how well the similarity of given pair propagates to the neighbours.

Structure Level Approaches

Approaches that considers the ontology in graph structure and consider upper and lower levels of the given concepts to find out similarity among concepts of different Ontologies.

Structural Similarity

COMA(Aumueller et al., 2005) system employs path similarity as a basis for calculating similarity among concept. SMART tool employs algorithm that considers structure of the relation in the vicinity of the concept being processed. It is implemented with PROMPT system that can be plugged in with Protégé-a widely accepted open-source knowledge engineering tool. Anchor-PROMPT(Noy & Musen, 2001) extends the simple PROMPT by calculating similarity measures based on ontology structure.

Semantic Matching

S-Match focuses on computing semantic matching between graph structures extracted from separate Ontologies. With this technique it is possible to differentiate the meaning of Concept of node and Concept at node.

Lattice

FCA-Merge(Stumme, 2005) incorporates machine learning Technique to derive a lattice which is then used to derive merged Ontology. Documents are accepted as inputs that provides the concepts to build the Ontology.

Machine Learning and Statistics Based Approach

While determination of exact mapping achieved by establishing string and structure level similarities, the performance can be further improved by employing methods to approximate the nearness. Machine learning, probability and statistics techniques can be incorporating in mapping techniques to improve the performance and achieving automation in matching process. GLUE(Doan et al., 2002) employs an instance based machine learning technique to semi-automatically create mappings for input Ontologies. Concept

D

Similarity in terms of joint probability distribution of instances of given concept. Ontology Mapping Enhancer (OMEN) (Mitra et al., 2005) adopts generation of Bayesian Net on the basis of mappings defined a priori. ITTalk (Cost et al., 2002) adopts Bayesian reasoning along with text based classification technology that collects similarity information in source Ontologies. The discovered semantic similarity is codified as labels and arcs of a graph.

Community Oriented Approach

CAIMAN(Lacher & Groh, 2001) proposed a scenario where member of communities want to express their viewpoints on categorization in community repository. COMA++(Aumueller et al., 2005) offers community driven ontology mapping by providing support for web based acquisition and sharing of ontology mapping.

Logic Based Approach

Description Logic (DL) is widely accepted for knowledge representation task. It is observed that Ontology creation is commonly carried out with commonly used tools that supports OWL-DL representation and reasoning. The mapping language that is selected to represent

mapping can also benefit from the representation and reasoning capability of DL. MAFRA(Maedche et al., 2002) uses DL for representing Semantic Bridge Ontology (SBO). OntoMapO uses DL for creating a Meta Ontology for consistent representation of Ontologies and mapping among them. CTXMatch and S-Match employs satisfiability techniques that is achieved using DL. ConcepTool(Compatangelo & Meisel, 2003) system uses DL in formalizing class-centered enhanced entity relationship model.

FUTURE TRENDS

Organizations are increasingly adopting knowledge-based approaches in building systems. The Ontology mapping techniques discussed here provides overview of efforts that enables integration of multiple Ontologies in the context of targeted application. Along with the syntax and standardized content, it will become necessary to consider Ontologies with various levels of detail to be mapped for complete and accurate coverage. Extending the current approaches that take a few Ontologies as input selected from vast pool based on the availability of required elements or structures, it will be necessary to consider specific types of Ontol-

Figure 1. Ontology mediation requirement in future applications
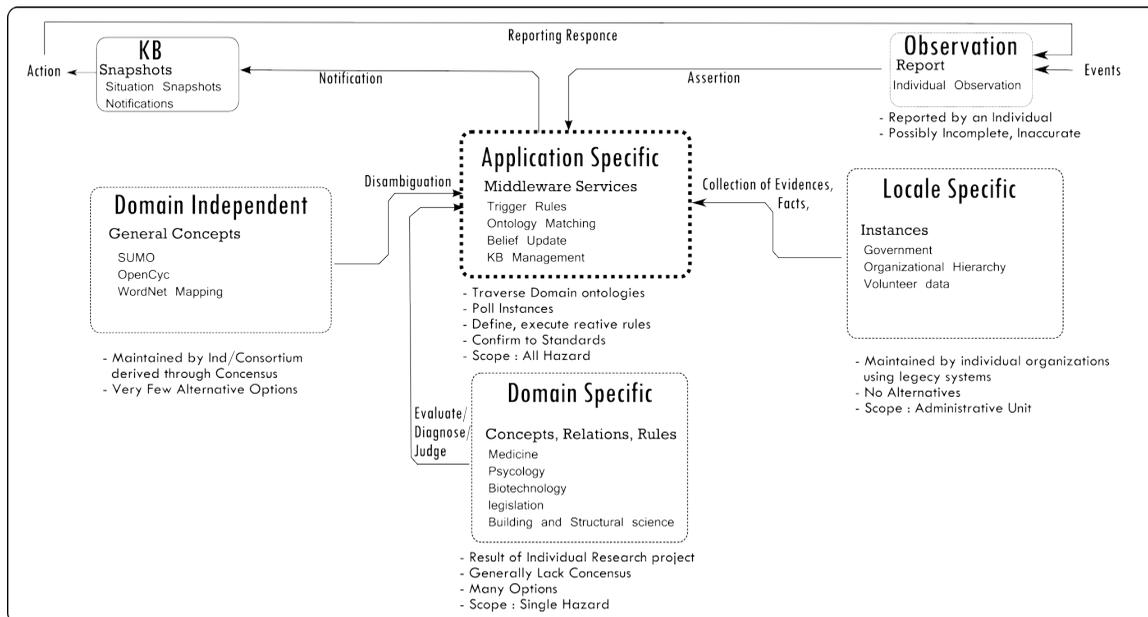
ogy. Figure 1 indicates one of the challenging open problems of investigating appropriate methods that allows integration of Ontologies defining Domain Independent, Domain Specific, Local Specific and Application Specific concepts to provide complete coverage of knowledge representation. This approach ensures that knowledge engineers can reuse the integrated part of Domain Independent and Domain Specific concepts and focus on only local specific concept to suitably integrate with application being built. For explaining a example scenario Figure 1 indicates the Ontology integration required for Disaster Management Agencies across the world. The Domain Specific and Domain Independent concepts are required by every agency and can be directly integrated with local specific and application specific concepts that can be unique to each implementing agency.

CONCLUSION

With proliferation of knowledge representation and reasoning techniques and standard based tools, domain knowledge captured in the form of Ontology is increasingly being available for reuse. The quality and quantity and level of detail with which domain concepts are defined differ considerably based on the discretion of knowledge engineer. The reuse requires concepts defined in multiple such Ontologies to be extracted or mapped to a resulting comprehensive representation that suits the requirement of problem on hand. This in-turn introduce the problem of syntactic and semantic heterogeneity. This article provided state-of-the-art in present techniques targeted at resolving heterogeneity.

REFERENCES

Antoniou, G., & Harmelen, F. van. (2004). Web ontology language: Owl. In S. Staab & R. Studer (Eds.), *Handbook on ontologies* (p. 67-92). Springer.

Aumueller, D., Do, H.-H., Massmann, S., & Rahm, E. (2005). Schema and ontology matching with coma++. *In Sigmod '05: Proceedings of the 2005 acm sigmod international conference on management of data* (pp. 906–908). NY, USA: ACM Press.

Brachman, R. J., & Levesque, H. J. (2004). *Knowledge representation and reasoning*. Morgan Kaufmann Publishers.

Choi, N., Song, I.-Y., & Han, H. (2006). A survey on ontology mapping. *SIGMOD Rec., 35 (3)*, 34–41.

Compatangelo, E., & Meisel, H. (2003). Conceptool: Intelligent support to the management of domain knowledge. In V. Palade et al. (Eds.), *Kes* (Vol. 2773, p. 81-88). Springer.

Cost, R. S., Finin, T. W., Joshi, A., Peng, Y., Nicholas, C. K., Soboroff, I., et al. (2002). Ittalks: A case study in the semantic web and daml+oil. *IEEE Intelligent Systems*, 17 (1), 40-47.

Davies, J., Studer, R., & Warren, P. (2006). *Semantic web technologies trends and research in ontology-based systems*. John Wiley & Sons Ltd.

Dhamankar, R., Lee, Y., Doan, A., Halevy, A., & Domingos, P. (2004). imap: discovering complex semantic matches between database schemas. In *Sigmod '04: Proceedings of the 2004 acm sigmod international conference on management of data* (pp. 383–394). NY, USA: ACM Press.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., et al. (2004). Swoogle: a search and metadata engine for the semantic web. In Cikm '04: *Proceedings of the thirteenth acm international conference on information and knowledge management* (pp. 652–659). NY, USA: ACM Press.

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2002). Learning to map between ontologies on the semantic web. In Www '02: *Proceedings of the 11th international conference on world wide web* (pp. 662–673). NY, USA: ACM Press.

Ehrig, M., & Sure, Y. (2005). Foam - framework for ontology alignment and mapping - results of the ontology alignment evaluation initiative. In B. Ashpole, M. Ehrig, J. Euzenat, & H. Stuckenschmidt (Eds.), *Integrating ontologies* (Vol. 156). CEUR-WS.org.

Ehring, M. (2007). *Ontology alignment bridging the semantic gap*. Springer.

Euzenat, J., Gu´egan, P., & Valtchev, P. (2005). Ola in the oaei 2005 alignment contest. In B. Ashpole et al. (Eds.), *Integrating ontologies* (Vol.156). CEUR-WS.org.

Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2005). S-match: an algorithm and an implementation of semantic matching. In Y. Kalfoglou et al. (Eds.), *Se-*

mantic interoperability and integration (Vol. 04391). IBFI, Germany.

Horrocks, I., & Patel-Schneider, P. F. (2001). The generation of daml+oil. In *Description logics*.

Kalfoglou, Y., Hu, B., Reynolds, D., & Shadbolt, N. (2005). *Semantic integration technologies survey* (Tech. Rep. No. 10842). University of Southampton.

Kalfoglou, Y., & Schorlemmer, W. M. (2005). Ontology mapping: The state of the art. In Y. Kalfoglou et al. (Eds.), *Semantic interoperability and integration* (Vol. 04391). IBFI, Germany.

Kotis, K., Vouros, G. A., & Stergiou, K. (2006). Towards automatic merging of domain ontologies: The hcone-merge approach. *J. Web Sem.*, 4 (1), 60-79.

Lacher, M. S., & Groh, G. (2001). Enabling personal perspectives on heterogeneous information spaces. *wetice*, 00, 285.

Le, B. T., Dieng-Kuntz, R., & Gandon, F. (2004). On ontology matching problems - for building a corporate semantic web in a multi-communities organization. In *Iceis* (4) (p. 236-243).

Li, J. (2004). *Lom: A lexicon-based ontology mapping tool.* Teknowledge Corporation, Palo Alto, CA.

Madhavan, J., Bernstein, P. A., & Rahm, E. (2001). Generic schema matching with cupid. In P. M. G. Apers et al. (Eds.), *Vldb* (p. 49-58). Morgan Kaufmann.

Maedche, A., Motik, B., Silva, N., & Volz, R. (2002). Mafra - a mapping framework for distributed ontologies. In A. G´omez-P´erez & V. R. Benjamins (Eds.), *Ekaw* (Vol. 2473, p. 235-250). Springer.

Manola, F., & Miller, E. (2004, February). *Rdf primer* (W3C Recommendation). World Wide Web Consortium.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Commun. ACM*, 38 (11), 39–41.

Mitra, P., Noy, N. F., & Jaiswal, A. (2005). Omen: A probabilistic ontology mapping tool. In Y. Gil, et al. (Eds.), *International semantic web conference* (Vol.3729, p. 537-547). Springer.

Niles, I., & Pease, A. (2001). Towards a standard upper ontology. *In Fois '01: Proceedings of the international conference on formal ontology in information systems* (pp. 2–9). NY, USA: ACM Press.

Niles, I., & Pease, A. (2003). Linking lixicons and ontologies: Mapping wordnet to the suggested upper merged ontology. In H. R. Arabnia (Ed.), *Ike* (p. 412-416). CSREA Press.

Noy, N., & Musen, M. (2000). Prompt: Algorithm and tool for automated ontology merging and alignment. In *Aaai/iaai* (p. 450-455). AAAI Press / The MIT Press.

Noy, N., & Musen, M. (2001). Anchor-prompt: Using non-local context for semantic matching. *In Workshop on ontologies and information sharing* (ijcai-2001), WA.

Noy, N. F. (2004a). Semantic integration: A survey of ontology based approaches. ACM SIGMOD *Record*, 33 (4), 65-70.

Noy, N. F. (2004b). Tools for mapping and merging ontologies. In S. Staab & R. Studer (Eds.), *Handbook on ontologies* (p. 365-384). Springer.

Predoiu, L., Feier, C., Scharffe, F., Bruijn, J. de, Martin-Recuerda, F., Manov, D., et al. (2006). *State-of-the-art survey on ontology merging and aligning v2,* (Tech. Rep.). DERI, University of Innsbruck.

Raskin, R. (2003). Semantic web for earth and environmental terminology (sweet). In Proc. of *nasa earth science technology conference*.

Shvaiko, P., & Euzenat, J. (2006). *A survey of schema-based matching approaches* (Tech. Rep.). INRIA, France,.

Sicilia, M.- ´A., Garcia, E., Sanchez, S., & Rodriguez, E. (2004). On integrating learning object metadata inside the opencyc knowledge base. In *Icalt*.

Stumme, G. (2005). Ontology merging with formal concept analysis. In Y. Kalfoglou et al. (Eds.), *Semantic interoperability and integration* (Vol. 04391). IBFI, Germany.

KEY TERMS

Articulation Ontology: Articulation ontology consists of concepts and relations that are identified

as link among the concepts defined in two separate Ontologies also known as articulation rules.

Context Aware Techniques: Techniques that are focused on nearness of weight assigned to specific relations among concepts considering application context as basis for mapping.

Extension Aware Techniques: Techniques that are focused on finding nearness among features of available instances of different Ontologies to form basis for mapping.

Intension Aware Techniques: Based on the Information flow theory, techniques that are focused on finding two different tokens (instances) belonging to separate Ontologies that maps to single type (concept) as a basis for mapping.

Linguistic Similarity Techniques: Set of techniques that refer linguistic nearness of concepts in the form of synonyms, hypernyms, and hyponyms by referring to related entries in the thesaurus as basis for mapping.

Ontology Alignment: Ontology Alignment is a process to articulate similarity in the form of one-to-one equality relation between every elements of two separate Ontologies.

Ontology Integration: Ontology Integration is a process that results in generation of a new ontology derived as a union of two or more source Ontologies of different but related subject domain.

Ontology Mapping: Ontology Mapping is a process to articulate similarities among the concepts belonging to separate source Ontologies.

Ontology Mediation: Ontology mediation is a process that reconciles difference between separate Ontologies to achieve semantic interoperability by performing alignment, mapping, merging and other required operations.

Ontology Merging: Ontology Mapping is a process that results in generation of a new ontology derived as a union of two or more source Ontologies of same subject domain.

Semantic Similarity Techniques: Techniques that are focused on logic satisfiability as basis of mapping.

String Similarity Techniques: Set of techniques that uses syntactic similarity of concepts as basis of mapping.

Structure Aware Techniques: Techniques that also consider structural hierarchy of concepts as basis of mapping.

Disk-Based Search

D

Stefan Edelkamp
University of Dortmund, Germany

Shahid Jabbar
University of Dortmund, Germany

INTRODUCTION

The need to deal with large data sets is at the heart of many real-world problems. In many organizations the data size has already surpassed Petabytes (10^{15}). It is clear that to process such an enormous amount of data, the physical limitations of RAM is a major hurdle. However, the media that can hold huge data sets, i.e., hard disks, are about a 10,000 to 1,000,000 times slower to access than RAM. On the other hand, the costs for large amounts of disk space have considerably decreased. This growing disparity has led to a rising attention to the design of *external memory algorithms* (Sanders et al., 2003) in recent years.

In a hard disk, random disk accesses are slow due to disk latency in moving the head on top of the data. But once the head is at its proper position, data can be read very rapidly. External memory algorithms exploit this fact by processing the data in the form of blocks. They are more informed about the future accesses to the data and can organize their execution to have minimum number of block accesses.

Traditional graph search algorithms perform well as long as the graph can fit into the RAM. But for large graphs these algorithms are destined to fail. In the following, we will review some of the advances in the field of search algorithms designed for large graphs.

BACKGROUND

Most modern operating systems provide a general-purpose memory management scheme called *Virtual Memory* to compensate for the limited RAM. Unfortunately, such schemes pay off only when the algorithm's memory accesses are local, i.e., it works on a particular memory address range for a while, before switching the attention to another range. Search algorithms, especially those that order the nodes on some particular node property, do not show such behaviour. They jump back and forth to pick the best node, in a spatially unrelated way for only marginal differences in the node property.

External memory algorithms are designed with a hierarchy of memories in mind. They are analyzed on an *external memory model* as opposed to the traditional von Neumann RAM model. We use the two-level memory model by Vitter and Shriver (1994) to describe the search algorithms. The model provides the necessary tools to analyze the asymptotic number of block accesses (I/O operations) as the input size grows. It consists of

- *M*: Size of the internal memory in terms of the number of elements,
- $N \gg M$: Size of the input in terms of the number of elements, and
- *B*: Size of the data block that can be transferred between the internal memory and the hard disk; transferring one such block is called as a single I/O operation.

The complexity of external memory algorithms is conveniently expressed in terms of predefined I/O operations, such as, *scan*(*N*) for scanning a file of size *N* with a complexity of $\Theta(N/B)$ I/Os, and *sort*(*N*) for external sorting a file of size *N* with a complexity of $\Theta(N/B \log_{M/B} (N/B))$ I/Os. With additional parameters the model can accommodate multiple disks and multiple processors too.

In the following, we assume a graph as a tuple (*V*, *E*, *c*), where *V* is the set of nodes, *E* the set of edges, and *c* the weight function that assigns a non-zero positive integer to each edge. If all edges have the same weight, the component *c* can be dropped and the graphs are called as *unweighted*. Given a start node s and a goal node g, we require the search algorithm to return an optimal path wrt. the weight function.

EXTERNAL MEMORY SEARCH ALGORITHMS

External Memory Breadth-First Search

Breadth-first search (BFS) is one of the basic search algorithms. It explores a graph by first expanding the nodes that are closest to the start node. BFS for external memory has been proposed by Munagala and Ranade (1999). It only considers undirected and explicit (provided beforehand in the form of adjacency lists) graphs. The working of the algorithm is illustrated on a graph in Fig. 1. Let *Open*(*i*) be the set of nodes at BFS level i residing on disk. The algorithm builds *Open*(*i*) from *Open*(*i*-1) as follows. Let *Succ*(*Open*(*i*-1)) be the multi-set of successors of nodes in *Open*(*i*-1); this set is created by concatenating all adjacency lists of nodes in *Open*(*i*-1). As there can be multiple copies of the same node in this set the next step is to remove these duplicate nodes. In an internal memory setting this can be done easily using a hash table. Unfortunately, in an external setting a hash-table is not affordable due to random accesses to its contents. Therefore, we rely on alternative methods of duplicates' removal that are well-suited for large data on disk. The first step is to sort the successor set using external sorting algorithms resulting in duplicate nodes lying adjacent to each

other. By an external scanning of this sorted set, all duplicates are removed. Still, there can be nodes in this set that have already been expanded in the previous layers. Munagala and Ranade proved that for undirected graphs, it is sufficient to subtract only two layers, *Open*(*i*–1) and *Open*(*i*–2), from *Open*(*i*). Since all three lists are sorted, this can be done by a parallel external scanning. The accumulated I/O complexity of this algorithm is $O(|V| + sort(|E|))$ I/Os, where $|V|$ is for the unstructured access to the adjacency lists, and $sort(|E|)$ for duplicates removal.

An implicit graph variant of the above algorithm has been proposed by Korf (2003). It applies $O(sort(|Succ(Open(i-1))|) + scan(|Open(i-1)| + |Open(i-2)|))$ I/Os in each iteration. Since no explicit access to the adjacency list is needed (as the state space is generated on-the-fly), by using $\Sigma_i |Succ(Open(i))| = O(|E|)$ and $\Sigma_i |Open(i)| = O(|V|)$, the total execution time is bounded by $O(sort(|E|) + scan(|V|))$ I/Os.

To reconstruct a solution path, we may store predecessor information with each node on disk (thus doubling the state vector size). Starting from the goal node, we recursively search for its predecessor in the previous layer through external scanning. The process continues until the first layer containing the start node is reached. Since the Breadth-first search preserves the shortest paths in a uniformly weighted graph, the

Figure 1. An example graph (left); Stages of External Breadth-First Search (right). Each horizontal bar corresponds to a file. The grey-shaded A(2) and A'(2) are temporary files.
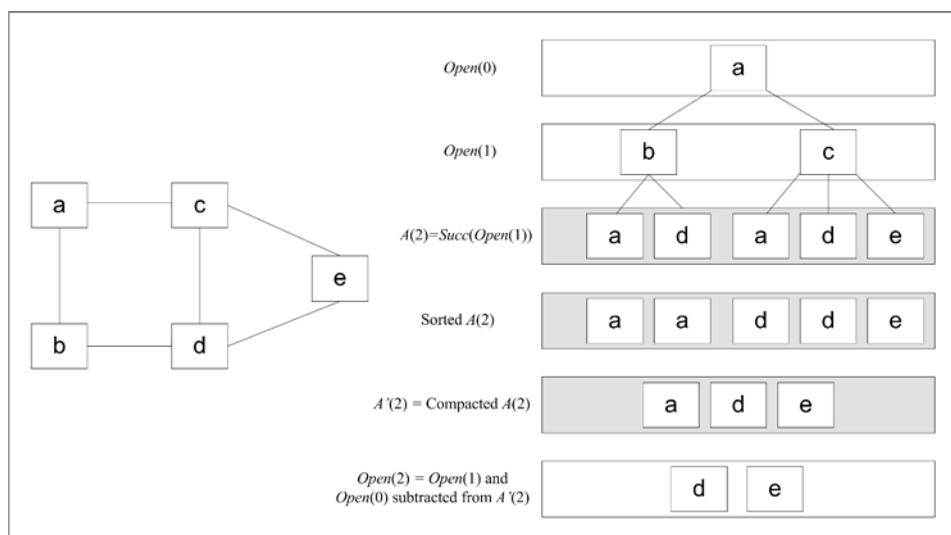
constructed path is the optimal one. The complexity is bounded by the scanning time of all layers in consideration, i.e., by $O(scan(|V|))$ I/Os.

External Memory Heuristic Search

Heuristic search algorithms utilize some form of guidance; be it user-provided or automatically inferred from the problem structure, to hone in on the goal. In practice, such algorithms are very effective when compared with the blind search algorithms like BFS. A* (Hart et al., 1968) is one such algorithm that prioritizes the nodes based on their actual distance from the start node along with their heuristic estimate to the goal state. Formally, if $g(n)$ represents the path cost for reaching a node n, and $h(n)$ the heuristic estimate of reaching the goal starting from n, then A* orders the nodes based on their f-value defined as $f(n)=g(n)+h(n)$. For an efficient implementation of A*, a priority queue data structure is required that allows us to remove the node with the minimum f-value for expansion. If the heuristic function h is *consistent*, then on each search path, no successor will have a smaller f-value than the predecessor. Therefore, A* – traversing the node set in f-order – expands each node at most once.

External A * by Edelkamp et al. (2004) maintains the search frontier on disk as files. Each file corresponds to an external representation of a bucket-based priority queue data structure. A *bucket* is a set of nodes sharing common properties. In a 1-level bucket implementation of A* by Dial (1969) each bucket addressed with index i contains all nodes u that have priority $f(n)=i$. A refinement proposed by Jensen et al. (2002) distinguishes between nodes with different g-values, and designates bucket $Open(i,j)$ to all nodes n with path length $g(n)=i$ and heuristic estimate $h(n)=j$. An external memory representation of this data structure memorizes each bucket in a different file. During the exploration process, only nodes from $Open(i,j)$ with $i+j=f$ are expanded, up to its exhaustion. Buckets are selected in lexicographic order for (i,j). By that time, the buckets $Open(i',j')$ with $i'<i$ and $i'+j'=f$ are *closed*, whereas the buckets $Open(i',j')$ with $i'+j'>f$ or with $i'>i$ and $i'+j'=f$ are *open*. Depending on the expansion progress, nodes in the active bucket are either *open* or *closed*.

It is practical to pre-sort buffers in one bucket immediately by an efficient internal sorting algorithm to ease merging. Duplicates within an active bucket are eliminated by merging all the pre-sorted buffers cor-

responding to the same bucket, resulting in one sorted file. This file can then be scanned to remove the duplicate nodes from it. In fact, both the merging and removal of duplicates can be done simultaneously. Another case of the duplicate nodes appears, when the nodes that have already been evaluated in the upper layers are generated again. As in the algorithm of Munagala and Ranade, External A* exploits the observation that, in an undirected problem graph, duplicates of a node with BFS-level i can at most occur in levels i, i–1 and i–2. In addition, since h is a total function, we have $h(n) = h(n')$, if $n = n'$. These duplicate nodes can be removed by file subtraction for the next active bucket $Open(g+1,h-1)$. We remove any node that has appeared in buckets $Open(g,h-1)$ and $Open(g-1,h-1)$. This file subtraction can be done by a mere parallel scan of the pre-sorted files and by using a temporary file in which the intermediate result is stored. It suffices to perform the duplicate removal only for the bucket that is to be expanded next.

Duplicate Detection Scope

The number of previous layers that are sufficient for full duplicate detection in directed graphs, is dependent on a property of the search graph called *locality* (Zhou and Hansen, 2006). In the following, we generalize their concept to weighted and directed search graphs. For a problem graph with node set V, discrete cost function c, successor set *Succ*, initial state s, and δ being defined as the minimal cost between two states, the shortest-path locality is defined as $L = \max\{\delta(s,n) - \delta(s,n') + c(n,n') \mid n, n' \in V, n' \in Succ(n)\}$. In unweighted graphs, we have $c(n,n')=1$ for all n, n'. Moreover, $\delta(s,n)$ and $\delta(s,n')$ differ by at most 1, so that the locality is 2, which is consistent with the observation of Munagala and Ranade.

The locality determines the *thickness* of the search frontier needed to prevent duplicates from appearing in the search. While the locality is dependent on the graph, the duplicate detection scope also depends on the search algorithm applied. For BFS, the search tree is generated with increasing path lengths (number of edges), while for weighted graphs the search tree is generated with increasing path cost (this corresponds to Dijkstra's algorithm in the one-level bucket priority queue data structure).

In a positively weighted search graph, the number of buckets that need to be retained to prevent duplicate

search effort is equal to the shortest-path locality of the search graph. Let us consider two nodes *n* and *n'*, with $n' \in Succ(n)$. Assume that n has been expanded for the first time, generating the successor *n'* which has already appeared in the layers 0, ..., $\delta(s,n)$–*L* implying $\delta(s,n') \leq \delta(s,n) - L$. We have, $L \geq \delta(s,n) - \delta(s,n') + c(n,n') \geq \delta(s,n) - (\delta(s,n) - L) + c(n,n') = L + c(n,n')$, in contradiction to $c(n,n') > 0$.

Refinements

Improvements of Munagala and Ranade's algorithm for explicit undirected graphs have been proposed by Mehlhorn and Meyer (2002), where a more structured access to the adjacency lists has been proposed. Ajwani et al. (2007) present an extensive empirical comparison of these approaches on different kinds of graphs.

Hash-based delayed duplicate detection (Korf and Schultze, 2005) is designed to avoid the complexity of sorting. It is based on two orthogonal hash functions. The primary hash function distributes the nodes to different files. Once a file of successors has been generated, duplicates are eliminated. The assumption is that all nodes with the same primary hash address fit into main memory. The secondary hash function maps all duplicates to the same hash address.

Structured duplicate detection (Zhou and Hansen, 2004) builds up an abstract graph on top of the problem graph through a disjoint partitioning. For expansion, all states that are mapped to the same abstract node are loaded into the memory along with the nodes that are mapped to the neighbouring abstract nodes. The partition is defined in such a way that any two adjacent nodes in the graph are mapped either to the same partition or to a neighbouring abstract partition. The successor nodes are checked against the neighbouring partitions and are removed if found as duplicates, as soon as they are generated.

Applications

Implementations for external *model checking* algorithms have been proposed by Kristensen and Mailund (2003), who suggested a sweep-line technique for scanning the search space according to a given partial order. For general LTL (Linear Temporal Logic) model checking, Edelkamp and Jabbar (2006) have extended External A* for *safety* and *liveness* checking and integrated it into the state-of-the-art model checker, SPIN.

FUTURE TRENDS

It is often the case that external memory algorithms can be lifted to parallel algorithms. With the advent of multi-core processors and affordable PC clusters, parallel algorithms become more and more important. Jabbar and Edelkamp (2006) provide a parallel implementation of the External A* for model checking safety properties. For probabilistic and non-deterministic models *value iteration* (VI) has been extended by Edelkamp et al. (2007) to work on large state spaces that cannot fit into the RAM. Instead of working on states, it works on edges $(n,n',a,h(n'))$, where *n* is called the predecessor state, *n'* the stored state, a the action that transforms *n* into *n'*, and $h(n')$ is the current value for *n'*. Similarly to the internal version of VI, the external version of VI works in two phases. A forward phase, where the state space is generated, and a backward phase, where the heuristic values are repeatedly updated until an ε-optimal policy is computed, or a maximum iterations are performed.

CONCLUSION

We have presented a brief overview of disk-based search algorithms. These algorithms are especially designed for large graphs that cannot fit into the RAM. An external variant of one of the basic search algorithms, i.e., Breadth-first search has been introduced. For the domains where a search algorithm can be equipped with some form of guidance to reduce the search efforts, External A* provides a complete and I/O efficient extension of the famous A* search algorithm. The whole paradigm of disk-based search is largely dependant on alternate forms of duplicate detection schemes. The most general one is sorting-based delayed duplicate detection. For special problems where a good disjoint partitioning of the graph is possible, hash-based duplicate detection and structured duplicated detection are feasible choices. We have also presented a generalization of the duplicate detection scope that dictates the number of previous layers that have to be checked to guarantee that no node will be expanded twice. Finally, we saw some future trends directed towards an efficient utilization of modern multi-core hardware and to policy search methods.

STXXL (Dementiev et al., 2005) provides an efficient library for external memory data structures and algorithms.

REFERENCES

Aggarwal, A., & Vitter, J.C. (1988). The *input/output complexity of sorting and related problems.* Journal of the ACM, 31(9):1116–1127.

Dial, R.B. (1969). Shortest-path forest with topological ordering. Communications of the ACM, 12(11):632–633.

Dementiev, R., Kettner, L., & Sanders, P. (2005). STXXL: Standard template library for XXL data sets. European Symposium on Algorithms (ESA), LNCS, 3669:640–651.

Ajwani, D., Meyer, U., & Osipov, V. (2007). *Improved External Memory BFS Implementation.* SIAM Workshop on Algorithm Engineering and Experiments (ALENEX), 3–12.

Edelkamp, S., Jabbar, S., & Schrödl, S. (2004). *External A*.* German Conference on Artificial Intelligence (KI), LNAI, 3238:233–250.

Edelkamp, S., & Jabbar, S. (2005). *Large-scale directed model checking LTL.* Model Checking Software (SPIN), LNCS, 3925:1–18.

Edelkamp, S., Jabbar, S., & Bonet, B. (2007). *External Memory Value Iteration.* International Conference on Automated Planning and Scheduling (ICAPS), *To appear*.

Hart, P.E., Nilsson, N.J., & Raphael, B. (1968). *A formal basis for heuristic determination of minimum path cost*. IEEE Transactions on Systems Science and Cybernetics, 4:100–107.

Jabbar, S. & Edelkamp, S. (2006). *Parallel external directed model checking with linear I/O.* Conference on Verification, Model Checking and Abstract Interpretation (VMCAI), LNCS, 3855:237–252.

Jensen, R.M., Bryant, R.E., & Veloso, M.M. (2002). SetA*: An efficient BDD-based heuristic search algorithm. National Conference on Artificial Intelligence (AAAI), 668–673.

Korf, R.E. (2003). Breadth-first frontier search with delayed duplicate detection. Model Checking and Artificial Intelligence (MOCHART), 87–92.

Korf, R.E., & Schultze, T. (2005) *Large-scale parallel breadth-first search*. National Conference on Artificial Intelligence (AAAI), 1380–1385.

Kristensen, L., & Mailund, T. (2003). Path finding with the sweep-line method using external storage. International Conference on Formal Engineering Methods (ICFEM), LNCS, 2885:319–337.

Mehlhorn, K., & Meyer, U. (2002). External-memory breadth-first search with sub-linear I/O. European Symposium on Algorithms (ESA), LNCS, 2461:723–735.

Munagala, K., & Ranade, A. (1999). I/O-complexity of graph algorithms. Symposium on Discrete Algorithms (SODA), 687–694.

Sanders, P., Meyer, U., & Sibeyn, J.F. (2003), Algorithms for Memory Hierarchies. Springer, LNCS, 2625.

Vitter, J.S., and Shriver, E.A.M. (1994). Algorithms for Parallel Memory I: Two-Level Memories. *Algorithmica*, 12(2-3):110–147.

Zhou, R. & Hansen, E. (2004). *Structured duplicate detection in external-memory graph search*. National Conference on Artificial Intelligence (AAAI), 683–689.

Zhou, R. & Hansen, E. (2006). *Breadth-First Heuristic Search*. Artificial Intelligence, 170(4–5):385–408.

KEY TERMS

Delayed Duplicate Detection: In difference to hash tables that eliminate duplicate states on-the-fly during the exploration, the process of duplicate detection can be delayed until a large set of states is available. It is very effective in external search, where it is efficiently achieved by external sorting and scanning.

Graph: A set of nodes connected through edges. The node at the head of an edge is called as the *target* and at the tail as the *source*. A graph can be *undirected*, i.e., it is always possible to return to the source through the same edge – the converse is a *directed* graph. If given beforehand in the form of adjacency lists (e.g., a road network), we call it an *explicit* graph. *Implicit* graphs – another name for 'state spaces' – are generated on-

the-fly from a start node and a set of rules/actions to generate the new states (e.g., a checkers game).

Heuristic Function: A function that assigns a node, an estimated distance to the goal node. For example, in route-planning, the Euclidean distance can be used as a heuristic function. A heuristic function is *admissible*, if it never overestimates the shortest path distance. It is also *consistent*, if it never decreases on any edge more than the edge weight, i.e., for a node n and its successor n', $h(n) - h(n') \leq c(n,n')$.

Memory Hierarchy: Modern hardware has a hierarchy of storage mediums: starting from the fast registers, the L1 and L2 caches, moving towards RAM and all the way to the slow hard disks and tapes. The latency timings on different levels differ considerably, e.g., registers: 2ns, cache: 20ns, hard disk: 10ms, tape: 1min.

Model Checking: It is an automated process that when given a model of a system and a property specification, checks if the property is satisfied by the system or not. The properties requiring that '*something bad will never happen*' are referred as *safety* properties, while the ones requiring that '*something good will eventually happen*' are referred as *liveness* properties.

Search Algorithm: An algorithm that when given two graph nodes, *start* and *goal*, returns a sequence of nodes that constitutes a path from *start* to the *goal*, if such a sequence exists. A search algorithm generates the successors of a node through an *expansion* process, after which, the node is termed as a *closed* node. The newly generated successors are checked for duplicates, and when found as unique, are added to the set of *open* nodes.

Value Iteration: Procedure that computes a policy (mapping from states to action) for a probabilistic or non-deterministic search problem most frequently in form of a Markov Decision Problem (MDP).

Distributed Constraint Reasoning

Marius C. Silaghi
Florida Institute of Technology, USA

Makoto Yokoo
Kyushu University, Japan

INTRODUCTION

Distributed constraint reasoning is concerned with modeling and solving naturally distributed problems. It has application to the coordination and negotiation between semi-cooperative agents, namely agents that want to achieve a common goal but would not give up private information over secret constraints. When compared to centralized constraint satisfaction (CSP) and constraint optimization (COP), one of the most expensive operations is communication. Other differences stem from new coherence and privacy needs. We review approaches based on asynchronous backtracking and depth-first search spanning trees.

Distributed constraint reasoning started as an outgrowth of research in constraints and multi-agent systems. Take the sensors network problem in Figure 1, defined by a set of geographically distributed sensors that have to track a set of mobile nodes. Each sensor can watch only a subset of its neighborhood

Figure 1. Sensor network
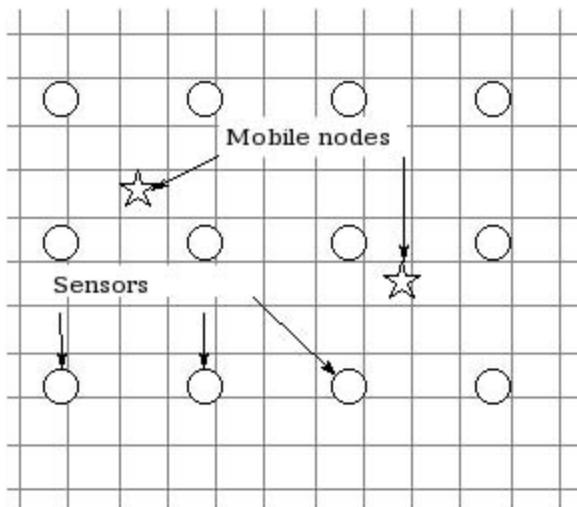
at a given time. Three sensors need to simultaneously focus on the same mobile node in order to locate it. Approaches modeling and solving this problem with distributed constraint reasoning are described in (Bejar, Domshlak, Fernandez, Gomes, Krishnamachari, Selman, &Valls, 2005).

There are two large classes of distributed constraint problems. The first class is described by a set of *Boolean relations* (aka *constraints*) on possible assignments of variables, where the relations are distributed among agents. They are called distributed constraint satisfaction problems (DisCSPs). The challenge is to find assignments of variables to values such that all these relations are satisfied. However, the reasoning process has to be performed by collaboration among the agents. There exist several solutions to a problem, and ties have to be broken by some priority scheme. Such priorities may be imposed from the problem description where some agents, such as government agencies, are more important than others. In other problems it is important to ensure that different solutions or participants have equal chances, and this property is called *uniformity*. When no solution exists, one may still want to find an assignment of the variables that conflict as few constraints as possible. The second class of problems refers to numerical optimization described by a set of functions (*weighted constraints*) defined on assignments of variables and returning positive numerical values. The goal is to find assignments that minimize the objective function defined by the sum of these functions. The problems obtained in this way are called distributed constraint optimization problems (DisCOPs). Some problems require a fair distribution of the amount of dissatisfaction among agents, minimizing the dissatisfaction of the most unsatisfied agent.

There are also two different ways of distributing a problem. The first way consists of distributing the data associated with it. It is defined in terms of which agents know which constraints. It can be shown that any

such problem can be translated into problems where all non-shared constraints are *unary* (constraints involving only one variable), also called *domain constraints*. Here one can assume that there exists a single unary constraint for each variable. It is due to the fact that any second unary constraint can be reformulated on a new variable, required to be equal to the original variable. The agent holding the unique domain constraint of a variable is called the *owner of that variable*. Due to the availability of this transformation many solutions focus on the case where only the unary constraints are not shared by everybody (also said to be private to the agents that know them). Another common simplification consists in assuming that each agent has a single unary constraint (i.e., a single variable). This simplification does not reduce the generality of the addressable problems since an agent can participate in a computation under several names, e.g., one instance for each unary constraint of the original agent. Such false identities for an agent are called pseudo-agents (Modi, Shen, Tambe, & Yokoo, 2005), or abstract agents (Silaghi & Faltings, 2005).

The second way of distributing a problem is in terms of who may propose instantiations of a variable. In such an approach each variable may be assigned a value solely by a subset of the agents while the other agents are only allowed to reject the proposed assignment. This distribution is similar to restrictions seen in some societies where only the parliament may propose a referendum while the rest of the citizens can only approve or reject it. Approaches often assume the simultaneous presence of both ways of distributing the problem. They commonly assume that the only agent that can make a proposal on a variable is the agent holding the sole unary constraint on that variable, namely its owner (Yokoo, Durfee, Ishida, & Kuwabara, 1998). When several agents are allowed to propose assignments of a variable, these authorized agents are called *modifiers* of that variable. An example is where each holder of a constraint on a variable is a legitimate modifier of that variable (Silaghi & Faltings, 2005).

BACKGROUND

The first challenge addressed was the development of *asynchronous algorithms* for solving distributed problems. Synchronization forces distributed processes to run at the speed of the slowest link. Algorithms that do not use synchronizations, namely where participants are at no point aware of the current state of other participants, are flexible but more difficult to design. With the exception of a few *solution detection* techniques (Yokoo & Hirayama, 2005), (Silaghi & Faltings, 2005), most approaches gather the answer to the problem by reading the state of agents after the system becomes idle and reaches the so called *quiescence* state (Yokoo et al., 1998). Algorithms that eventually reach quiescence are also called *self-stabilizing* (Collin, Dechter, & Katz, 1991). A *complete* algorithm is an algorithm that guarantees not to miss any existing solution. A *sound* algorithm is a technique that never *terminates* in a suboptimal state.

Another challenge picked by distributed constraint reasoning research consists of providing privacy for the sub-problems known by agents (Yokoo et al., 1998). The object of privacy can be of different types. The *existence of a constraint* between two variables may be secret as well as the *existence of a variable* itself. Many approaches only try to ensure the *secrecy of the constraints*, i.e., the hiding of the identity of the *valuations* that are penalized by that constraint. For optimization problems one also assumes a need to keep secret the amount of the penalty induced by the constraint. As mentioned previously, it is possible to model such problems in a way where all secret constraints are unary (approach known as having *private domains*). Some problems may have both secret and public constraints. Such public constraints may be used for an efficient preprocessing prior to the expensive negotiation implied by secret constraints. Solvers that support guarantees of privacy at any cost employ *cryptographic multi-party computations* (Yao 1982). There exist several cryptographic technologies for such computations, and some of them can be used interchangeably by distributed problem solvers. However, some of them offer information theoretical security guarantees (Shamir, 1979) being resistant to any amount of computation, while others offer only cryptographic security (Cramer, Damgaard, & Nielsen, 2000) and can be broken using large amounts of computation or quantum computers. The result of a computation may reveal secrets itself and its damages can be reduced by being careful in formulating the query to the solver. For example, less information is lost by requesting the solution to be picked randomly than by requesting the first solution. The computations can be done cryptographically by a group of *semi-trusted servers*,

or they can be performed by participants themselves. A third issue in solving distributed problems is raised by the size and the dynamism of the system.

DISTRIBUTED CONSTRAINT REASONING

Framework

A common definition of a distributed constraint optimization problem (DisCOP) (Modi et al., 2005) consists of a set of variables $X=\{x_1, ..., x_n\}$ and a set of agents $A=\{A_1, ..., A_n\}$, each agent A_i holding a set of constraints. Each variable x_i can be assigned only with those values which are allowed by a domain constraint D_i. A constraint Φ_j on a set of variables X_j is a function associating a positive numerical value to each combination of assignments to the variables X_j. The typical challenge is to find an assignment to the variables in X such that the sum of the values returned by the constraints of the agents is minimized.

A tuple of assignments is also called a *partial solution*. A restriction often used with DisCOPs requires that each agent A_i holds only constraints between xi and a subset of the previous variables, $\{x_1,...,x_{i-1}\}$. Also, for any agent A_i, the agents $\{A_1, ..., A_{i-1}\}$ are the predecessors of A_i and the agents $\{A_{i+1}, ..., A_n\}$, are its successors.

To understand the generality and limitations of this restriction, consider a conference organization problem with 3 variables x_1 (time), x_2 (place), and x_3 (general chair) and 3 constraints Φ_{12} (between x_1 and x_2), Φ_{23} (between x_2 and x_3), and Φ_{13} (between x_1 and x_3), where Alice has Φ_{12}, Bob enforces Φ_{23}, and Carol is interested in Φ_{13}, Figure 3.

This problem can be modeled as a DisCOP with 4 agents. Alice uses two agents, A_1 and A_2. The original participant is called physical agent and the agents of the model are called pseudo-agents. Bob uses the agent A_3 and Carol uses an agent A_4. The new variable x_4 of the agent A_4 is involved in a ternary constraint Φ_{134} with x_1 and x_3. The constraint Φ_{134} is constructed such that its projection on x_1 and x_2 is Φ_{13}.

However the restricted framework cannot help general purpose algorithms to learn and exploit the fact that agents A_1 and A_2 know each other's constraints. It also requires finding an optimal value for the variable x_4, which is irrelevant to the query. To avoid aforementioned limitations some approaches remove the restriction on which variables can be involved in the constraints of an agent and can obtain some improvements in speed

Figure 2. Translating between DisCOP frameworks
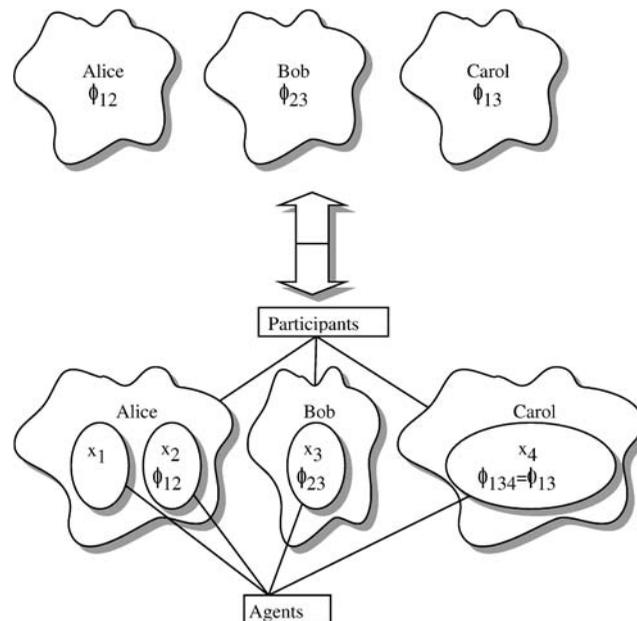
(Silaghi & Faltings, 2005). Other frameworks typically used with hill-climbing solvers, with solvers that reorder agents, and with arc consistency, assume that each agent A_i knows all the constraints that involve the variable x_i. This implies that any constraint between two variables x_i and x_j is known by both agents A_i and A_j. In general, a problem modeled as a DisCOP where any private constraint may be hold by any agent can be converted to its dual representation in order to obtain a model with this framework. When penalties for constraint violation can only take values in *{0,∞}*, corresponding to *{true, false}*, one obtains distributed constraint satisfaction problems.

A *protocol* is a set of rules about what messages may be exchanged by agents, when they may be sent, and what may be contained in their payload. A *distributed algorithm* is an implementation of a protocol as it specifies an exact sequence of operations to be performed as a response to each event, such as start of computation or receipt of a message. Autonomous self-interested agents are more realistically expected to implement protocols rather than to strictly adhere to algorithms. Protocols can be theoretically proved correct. However experimental validation and efficiency evaluation of a protocol is done by assuming that agents strictly follow some algorithm implementing that protocol.

Efficiency Metrics

The simplest metric for evaluating DisCOP solvers uses the time from the beginning of a distributed computation to its end. It is possible only with a real distributed system (or a very realistic simulation). The network load for benchmarks is evaluated by counting the total number of messages exchanged or the total number of bytes exchanged. The total time taken by a simulator yields the efficiency of a DisCOP solver when used as a weighted CSP solver. Another common metric is given by the *highest logic clocks* (Lamport, 1978) occurring during the computation. Lamport's logic clocks associate a cost with each message and another cost with each local computation. When the cost assigned to each message is 1 and the cost for local computations is 0, the obtained value gives the *longest sequential chain of causally ordered messages* (Silaghi & Faltings, 2005). When all message latencies are identical, this metric is equivalent to the number of rounds of a simulator where at each round an agent handles all messages received in the previous round (Yokoo et al., 1998).

If the cost assigned to each message is 0 and the cost of a constraint check is 1, the obtained value gives the number of *non-concurrent constraint checks* (NCCC) (Meisels, Kaplansky, Razgon, & Zivan, 2002). When a constraint check is assumed to cost a fraction of a message then the obtained value gives the *equivalent NCCCs (ENCCCs)*. One can evaluate the actual fraction between message latencies and constraint checks in the operating point (OP) of the target application (Silaghi & Yokoo, 2007). However many distributed solvers do not check constraints directly but via *nogoods* and there is no standardized way of accounting the handling of the latter ones.

Techniques

Solving algorithms span the range between full centralization, where all constraints are submitted to a central server that returns a solution, through incremental centralization (Mailler & Lesser, 2004), to very decentralized approaches (Walsh, Yokoo, Hirayama, & Wellman, 2003).

The Depth-First Search (DFS) spanning trees of the constraint graph proves useful for distributed DisCOP solvers. When used as a basis for ordering agents, the assignment of any node of the tree makes its subtrees independent (Collin, Dechter, & Katz, 2000). Such independence increases parallelism and decreases the complexity of the problem. The structure can be exploited in three ways. Subtrees can be explored in parallel for an opportunistic evaluation of the best branch, reminding of iterative A* (Modi et al., 2005). Alternatively a branch and bound approach can systematically evaluate different values of the root for each subtree (Chechetka & Sycara, 2006). A third approach uses dynamic programming to evaluate the DFS trees from leaves towards the root (Petcu & Faltings, 2006).

Asynchronous usage of *lookahead* techniques based on maintenance of *arc consistency* and *bound consistency* require handling of interacting data structures corresponding to different concurrent computations. Concurrent consistency achievement processes at different depths in the search tree have to be coordinated giving priority to computations at low depths in the tree (Silaghi & Faltings, 2005).

The concept at the basis of many asynchronous algorithms is the *nogood*, namely a self contained statement about a restriction to the valuations of the

variables, inferred from the problem. A *generalized valued nogood* has the form $[R,c,T]$ where T specifies a set of partial solutions $\{N_1,...,N_k\}$ for which the set of constraints R specifies a penalty of at least c. A common simplification, called *valued nogood* (Dago & Verfaille, 1996), refers to a single partial solution, $[R,c,N]$. Priority induced vector clock timestamps called *signatures* can be used to arbitrate between conflicting assignments concurrently proposed by several modifiers (Silaghi & Faltings, 2005). They can also handle other types of conflicting proposals, such as new ordering.

ADOPT-ing is an illustrative algorithm unifying the basic DisCSP and DisCOP solvers ABT (Yokoo et al., 1998) and ADOPT (Modi et al., 2005). It works by having each agent concurrently chose for its variable the best value given known assignments of *predecessors* and cost estimations received from *successors* (Silaghi & Yokoo 2007). Each agent announces its assignments to interested successors using **ok?** messages. Agents are interested in variables involved in their constraints or nogoods. When a nogood is received, agents announce new interests using **add-link** messages. A forest of DFS trees is dynamically built. Initially each agent is a tree, having no ancestors. When a constraint is first used, the agent adds its variables to his `ancestors` list and defines his *parent* in the DFS tree as the closest ancestor. Ancestors are announced of their own new ancestors. Nogoods inferred by an agent using resolution on its nogoods and constraints are sent to targeted predecessors and to its parent in the DFS tree using **nogood** messages, to guarantee optimality. Known costs of DFS subtrees for some values can be announced to those subtrees using **threshold** nogoods attached to **ok?** messages.

Example

An asynchronous algorithm could solve the problem in Figure 2 using the trace in Figure 3. In the messages of

Figure 3. The constraint graph of a DisCOP. The fact that the penalty associated with not satisfying the constraint $x_1 \neq x_2$ is 4, is denoted by the notation (#4).
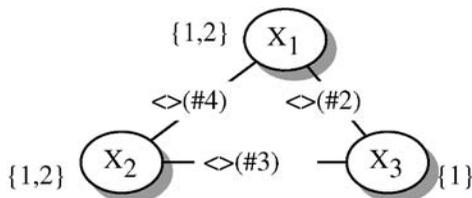
Figure 3, constraints are represented as Boolean values in an array. The ith value in this array set to T signifies that the constraints of A_i are used in the inference of that nogood. The agents start selecting values for their variables and announce them to interested lower priority agents. The first exchanged messages are **ok?** messages sent by A_1 to both successors A_2 and A_3 and proposing the assignment $x_1=1$. A_2 sends an **ok?** message to A_3 proposing $x_2=2$.

A_3 detects a conflict with x_1, inserts A_1 in its DFS tree ancestors list, and sends a nogood with cost 2 to A_1 (message 3). A_1 answers the received nogood by switching its assignment to a value with lower current estimated value, $x_1=2$ (message 4). A_2 reacts by switching x_2 to its lowest cost value, $x_2=1$ (message 5). A_3 detects a conflict with x_2 and inserts A_2 in its ancestors list, which becomes $\{A_1, A_2\}$. A_3 also announces the conflict to A_2 using the **nogood** message 6. This nogood received by A_2 is combined with the nogood locally inferred by A_2 for its value 2 due to the constraint $x_1 \neq x_2$(#4). That inference also prompts the insertion of A_1 in the ancestors list of A_2. The obtained nogood is therefore sent to A_1 using message 7. A_1 and later A_2 switch their assignments to the values with the lowest cost, attaching the latest nogoods received for those values as threshold nogoods (messages 8, 9 and 10). At this moment the system reaches *quiescence*.

FUTURE TRENDS

The main remaining challenges with distributed constraint reasoning are related to efficient ways of achieving privacy and with handling very large problems.

CONCLUSION

The distributed constraint reasoning paradigms allow easy specification of new problems. The notion varies largely between almost any two researchers. It can refer to the distribution of subproblems or it can refer to the distribution of authority in assigning variables. The reason and goals of the distribution vary as well, where either privacy of constraints, parallelism in computation, or size of data are cited as major concern. Most algorithms can be easily translated from one framework to the other, but they may not be appropriate for a new goal.

Figure 4. Simplified trace of an asynchronous solver (ADOPT-ing (Silaghi & Yokoo, 2007)) on the problem in Figure 3

$$
\begin{aligned}
&1.\ A_1 && \mathbf{ok?}\langle x_1, 1\rangle && \longrightarrow && A_2, A_3\\
&2.\ A_2 && \mathbf{ok?}\langle x_2, 2\rangle && \longrightarrow && A_3\\
&3.\ A_3 && \mathbf{nogood}([|F,F,T|, 2, \langle x_1,1\rangle], 3, \{A_1\}) && \longrightarrow && A_1\\
&4.\ A_1 && \mathbf{ok?}\langle x_1, 2\rangle && \longrightarrow && A_2, A_3\\
&5.\ A_2 && \mathbf{ok?}\langle x_2, 1\rangle && \longrightarrow && A_3\\
&6.\ A_3 && \mathbf{nogood}([|F,F,T|, 3, \langle x_2,1\rangle], 3, \{A_1,A_2\}) && \longrightarrow && A_2\\
&7.\ A_2 && \mathbf{nogood}([|F,T,T|, 3, \langle x_1,2\rangle], 2, \{A_1\}) && \longrightarrow && A_1\\
&8.\ A_1 && \mathbf{ok?}\langle x_1, 1\rangle && \longrightarrow && A_2\\
&9.\ A_1 && \mathbf{ok?}\langle x_1, 1\rangle \mathbf{threshold}\,[|F,F,T|, 2, \langle x_1,1\rangle] && \longrightarrow && A_3\\
&10.\ A_2 && \mathbf{ok?}\langle x_2, 2\rangle && \longrightarrow && A_3
\end{aligned}
$$

REFERENCES

Bejar, R., Domshlak, C., Fernandez, C., Gomes, C., Krishnamachari, B., Selman, B., &Valls, M. (2005). Sensor networks and distributed CSP: communication, computation and complexity. *Artificial Intelligence*, 161(1-2):117-147.

Chechetka, A., & Sycara, K. (2006). No-commitment branch and bound search for distributed constraint optimization. In *AAMAS*, 1427-1429.

Cramer, R., Damgaard, I., & Nielsen, J.B. (2000). *Multi-party Computation from Threshold Homomorphic Encryption*. BRICS RS-00-14.

Collin, Z.; and Dechter, R.; and Katz,S. (1991). On the feasibility of distributed constraint satisfaction. *IJCAI*, 318-324.

Collin, Z., Dechter, R., & Katz,S. (2000). Self-Stabilizing Distributed Constraint Satisfaction. *Chicago Journal of Theoretical Computer Science*, 3(4).

Dago, P., & Verfaillie, G. (1996). Nogood recording for valued constraint satisfaction problems. In *ICTAI*, 132-139

Lamport, L. (1978). Time, Clocks and the Ordering of Events in a Distributed System. *Communications of the ACM*. 21(7):558-565.

Mailler, R., & Lesser, V. (2004). Solving distributed constraint optimization problems using cooperative mediation. In *AAMAS*, 438-445.

Meisels, A., Kaplansky, E., Razgon, I., & Zivan, R. (2002). Comparing Performance of Distributed Constraints Processing Algorithms. *DCR*, 86-93.

Modi, P. J., Shen, W.-M., Tambe, M., & Yokoo, M. (2005). ADOPT: Asynchronous Distributed Constraint Optimization with Quality Guarantees. *Artificial Intelligence Journal* 161(1-2).

Petcu, A., & Faltings, B. (2006). ODPOP: an algorithm for open/distributed constraint optimization. In *AAAI*.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*. 22:612-613.

Silaghi, M.-C., & Faltings, B. (2005). Asynchronous aggregation and consistency in distributed constraint satisfaction. *Artificial Intelligence Journal* 161(1-2):25-53.

Silaghi, M.-C., & Yokoo, M. (2007). Dynamic DFS Tree in ADOPT-ing. *AAAI*.

Walsh, W.E., M. Yokoo,M., K. Hirayama, K., & M.P. Wellman, M.P. (2003). On market-inspired approaches to propositional satisfiability. *Artificial Intelligence*. 144: 125-156.

Walsh, T. (2007). Traffic light scheduling: a challenging distributed constraint optimization problem. In *DCR*.

Yao, A. (1982). Protocols for secure computations. *FOCS*. 160-164.

Yokoo, M., Durfee, E. H., Ishida, T., & Kuwabara, K. (1998). The Distributed Constraint Satisfaction Prob-

lem: Formalization and Algorithms. In *IEEE TKDE*, 10(5) 673-685.

Yokoo, M., & Hiramaya, K. (2005). The Distributed Breakout Algorithm. *Artificial Intelligence Journal*. 161(1-2), 229-246.

KEY TERMS

Agent: A participant in a distributed computation, having its own constraints.

Constraint: A relation between variables specifying a subset of their Cartesian product that is not permitted. Optionally it can also specify numeric penalties for those tuples.

DisCOP: Distributed Constraint Optimization Problem framework (also DCOP).

DisCSP: Distributed Constraint Satisfaction Problem framework (also DCSP).

Nogood: A logic statement about combinations of assignments that are penalized due to some constraints.

Optimality: The quality of an algorithm of returning only solutions that are at least as good as any other solution.

Quiescence: The state of being inactive. The system will not change without an external stimulus.

Distributed Representation of Compositional Structure

Simon D. Levy
Washington and Lee University, USA

INTRODUCTION

AI models are often categorized in terms of the connectionist vs. symbolic distinction. In addition to being descriptively unhelpful, these terms are also typically conflated with a host of issues that may have nothing to do with the commitments entailed by a particular model. A more useful distinction among cognitive representations asks whether they are *local* or *distributed* (van Gelder 1999).

Traditional symbol systems (grammar, predicate calculus) use local representations: a given symbol has no internal content and is located at a particular address in memory. Although well understood and successful in a number of domains, traditional representations suffer from brittleness. The number of possible items to be represented is fixed at some arbitrary hard limit, and a single corrupt memory location or broken pointer can wreck an entire structure.

In a distributed representation, on the other hand, each entity is represented by a pattern of activity distributed over many computing elements, and each computing element is involved in representing many different entities (Hinton 1984). Such representations have a number of properties that make them attractive for knowledge representation (McClelland, Rumelhart, & Hinton 1986): they are robust to noise, degrade gracefully, and support graded comparison through distance metrics. These properties enable fast associative memory and efficient comparison of entire structures without unpacking the structures into their component parts.

This article provides an overview of distributed representations, setting the approach in its historical context. The two essential operations necessary for building distributed representation of structures – *binding* and *bundling* – are described. We present example applications of each model, and conclude by discussing the current state of the art.

BACKGROUND

The invention of the backpropagation algorithm (Rumelhart, Hinton, & Williams 1986) led to a flurry of research in which neurally inspired models were applied to tasks for which the use of traditional AI data structures and algorithms were commonly assumed to be the only viable approach. A compelling feature of these new models was that they could "discover" the representations best suited to the modelling domain, unlike the manmade representations used in traditional AI. These discovered or learned representations were typically vectors of numbers in a fixed interval like [0, 1], representing the values of the hidden variables. A statistical technique like principal component analysis could be applied to such representations, revealing interesting regularities in the training data (Elman 1990).

Issues concerning the nature of the representations learned by backpropagation led to criticisms of this work. The most serious of these held that neural networks could not arrive at or exploit systematic, compositional representations of the sort used in traditional cognitive science and AI (Fodor & Pylyshyn 1988). A minimum requirement noted by critics was that a model that could represent e.g. the idea *John loves Mary* should also be able to represent *Mary loves John* (systematicity) and to represent *John*, *Mary*, and *loves* individually in the same way in both (compositionality). Critics claimed that neural networks are in principle unable to meet this requirement.

Systematicity and compositionality can be thought of as the outcome of two essential operations: *binding* and *bundling*. Binding associates fillers (*John*, *Mary*) with roles (lover, beloved). Bundling combines role/filler bindings to produce larger structures. Crucially, representations produced by binding and bundling must support an operation to recover the fillers of roles: it must be possible to ask "Who did what to whom?" questions and get the right answer. Starting

around 1990, several researchers began to focus their attention on building models that could perform these operations reliably.

VARIETIES OF DISTRIBUTED REPRESENTATION

This article describes the various approaches found in the recent neural network literature to implementing the binding and bundling operations. Although several different models have been developed, they fall into one of two broad categories, based on the way that roles are represented and how binding and bundling are performed.

Recursive Auto-Associative Memory

In Recursive Auto-Associative Memory, or RAAM (Pollack 1990), fillers are represented as relatively small vectors (N=10-50 elements) of zeros and ones. Roles are represented as $N \times N$ matrices of real values, and role/filler binding as the vector/matrix product. Bundling is performed by element-wise addition of the resulting vectors. There are typically two or three role matrices, representing general role categories like agent and patient, plus another $N \times N$ matrix must to represent the predicate (*loves*, *sees*, *knows*, etc.). Because all vectors are the same size N, vectors containing bindings can be used as fillers, supporting structures of potentially unlimited complexity (*Bill knows Fred said John loves Mary.*) The goal is to learn a set of matrix values (weights) to encode a set of such structures.

In order to recover the fillers, a corresponding set of matrices must be trained to *de*code the vectors produced by the encoder matrices. Together, the encoder and decoder matrices form an autoassociator network (Ackley, Hinton, & Sejnowski 1985) that can be trained with backpropagation. The only additional constraint needed for backprop is that the vector/matrix products be passed through a limiting function, like the sigmoidal "squashing" function $f(x) = 1 / (1 + e^{-x})$, whose output falls in the interval (0,1). Figure 1 shows an example of autoassociative learning for a simple hypothetical structure, using three roles, with $N = 4$. The same network is shown at different stages of training (sub-tree and full tree) during a single backprop epoch. Note that the network devises its own compositional representations on its intermediate ("hidden") layer, based on arbitrary binary vectors chosen by the experimenter. Unlike these binary vectors (black and white units), the intermediate representations can have values between zero and one (greyscale).

Once the RAAM network has learned a set of structures, the decoder sub-network should be able to recursively unpack each learned representation into its constituent elements. As shown in Figure 2, decoding is a recursive process that terminates when the decoder's output is similar enough to a binary string and continues otherwise. In the original RAAM formulation, "similar enough" was determined by thresholds: if a unit's value was above 0.8, it was considered to be on, and if it was below 0.2 it was considered to be off.

RAAM answered the challenge of showing how neural networks could represent compositional structures in a systematic way. The representations discovered by RAAM could be compared directly via distance metrics,

Figure 1. Learning the structure (knows bill (loves john mary))) with RAAM
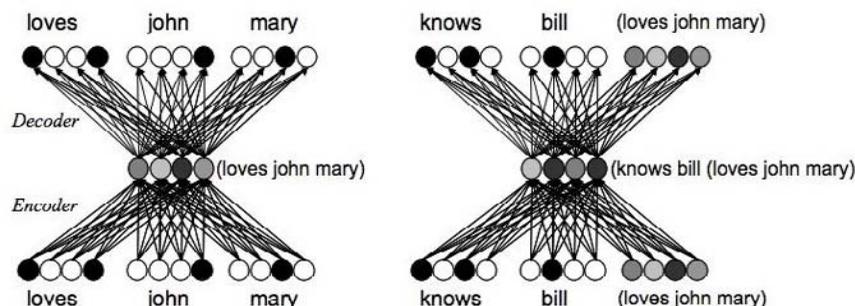
and transformed in a rule-like way, without having to recursively decompose the structure elements as in traditional localist models (Chalmers 1990). Nevertheless, the model failed to scale up reliably to data sets of more than a few dozen different structures. This limitation arose from the termination test, which created a variety of "halting problem": decoding often terminated too soon or continued indefinitely. In addition, encodings of novel structures were typically decoded to already-learned structures, a failure in generalization.

A number of solutions were developed to deal with this problem. One solution (Levy and Pollack 2001) built on the insight that the RAAM decoder is essentially an iterated function system, or IFS (Barnsley 1993). Use of the sigmoidal squashing function ensures that this IFS has an attractor, which is the infinite set (Cantor dust) of vectors reachable on an infinite number of feedback iterations from any initial vector input to the decoder. A more "natural" termination test is to check whether the output is a member of the set of vectors that make up this attractor. Fixing the numerical precision of the decoder results in a finite number of representable vectors, and a finite time to reach the attractor, so that membership in the attractor can be determined efficiently.

This approach to the termination test produced a RAAM decoder that could store a provably infinite number of related structures (Melnik, Levy, & Pollack 2001). Because the RAAM network was no longer an autoassociator, however, it was not clear what sort of algorithm could replace backpropagation for learning a specific, finite set of structures.

The other solution to the RAAM scaling problem discarded the nonlinear sigmoidal squashing function and replaced backprop with principal components analysis (PCA) as a means of learning internal representations (Callan 1996). This approach yielded the ability to learn a much larger set of structures in many fewer iterations (Voegtlin & Dominey 2005), and showed generalization similar in some respects to what has been observed for children acquiring a first language (Tomasello 1992).

Vector Symbolic Architectures

Vector Symbolic Architectures is a term coined by Gayler (2003) for a general class of distributed representation models that implement binding and bundling directly, without an iterative learning algorithm of the sort used by RAAM. These models can trace their origin to the Tensor Product model of Smolensky (1990). Tensor-product models represent both fillers and roles as vectors of binary or real-valued numbers. Binding is implemented by taking the tensor (outer) product of a role vector and a filler vector, resulting in a mathematical object (matrix) having one more dimension than the filler. Given vectors of sufficient length, each tensor product will be unique. As with RAAM, bundling can then be implemented as element-wise addition (Figure 3), and bundled structures can be used as roles, opening the door to recursion. To recover a filler (role) from a bundled tensor product representation, the product is simply divided by the role (filler) vector.

Because the dimension of the tensor product increases with each binding operation, this method suffers from the well-known "curse of dimensionality" (Bellman 1961). As more recursive embedding is performed, the size of the representation grows exponentially. The solution is to collapse the $N \times N$ role/filler matrix back into a length-N vector. As shown

Figure 2. Decoding a structure to its constituents
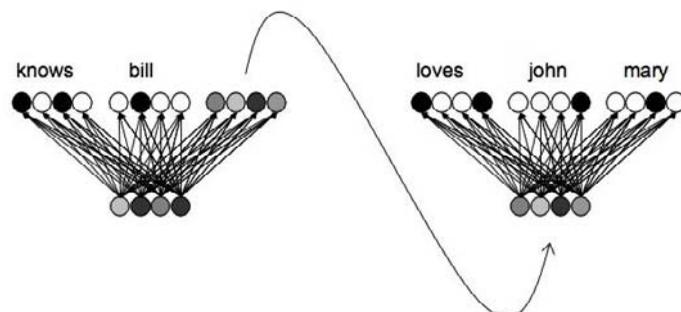
in Figure 4, there are two ways of doing this. In Binary Spatter Coding, or BSC (Kanerva 1994), only the elements along the main diagonal are kept, and the rest are discarded. If bit vectors are used, this operation is the same as taking the exclusive or (XOR) of the two vectors. In Holographic Reduced Representations, or HRR (Plate 1991), the sum of each diagonal is taken, with wraparound (circular convolution) keeping the length of all diagonals equal. Both approaches use very large ($N > 1000$ elements) vectors of random values drawn from a fixed set or interval.

Despite the size of the vectors, VSA approaches are computationally efficient, requiring no costly backpropagation or other iterative algorithm, and can be done in parallel. Even in a serial implementation, the BSC approach is $O(N)$ for a vector of length N, and the HRR approach can be implemented using the Fast Fourier Transform, which is $O(N \log N)$. The price paid is that most of the crucial operations (circular convolution, vector addition) are a form of lossy compression that introduces noise into the representations. The introduction of noise requires that the unbinding process employ a "cleanup memory" to restore the fillers to their original form. The cleanup memory can be implemented using Hebbian auto-association, like a Hopfield Network (Hopfield 1982) or Brain-State-in-a-Box model (Anderson, Silverstein, Ritz, & Jones 1977). In such models the original fillers are attractor basins in the network's dynamical state space. These methods can be simulated by using a table that stores the original vectors and returns the one closest to the noisy version.

Figure 3. Building a tensor product representation of John loves Mary
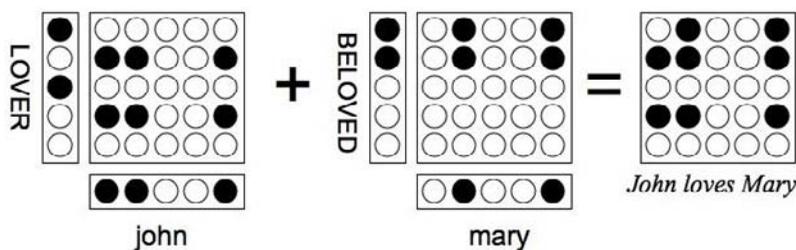
Figure 4. Methods for keeping fixed dimensionality in tensor-product representations
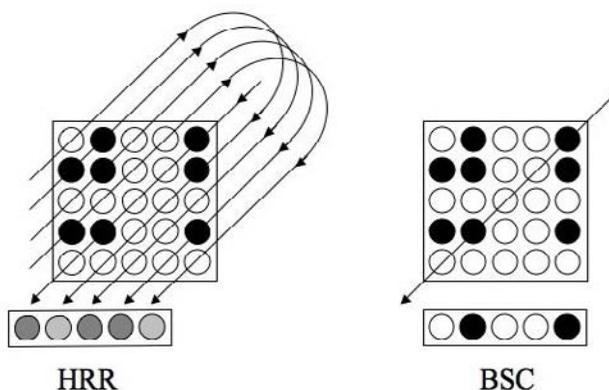
FUTURE TRENDS

Both the RAAM and VSA approaches have provided a basis for a significant amount of research in AI and cognitive science. The invention of the simplified linear RAAM, trained by relatively fast iterative method using principal components (Voegtlin & Dominey 2005) makes RAAM a practical approach to representing structure in distributed representation. Recent work in modelling language acquisition (Dominey, Hoen, & Inui 2006) hints at the possibility that RAAM may continue to thrive in its linear incarnation.

On the other hand, the simplicity of the VSA approach, using vector arithmetic operations available as primitives in many programming environments, has made it increasingly popular for modelling a variety of cognitive tasks – especially those requiring the integration of different types of information. For example, the DRAMA analogy model of Eliasmith & Thagard (2001) shows how HRR be used to integrate semantic and structural information in a way that closely parallels results seen in analogy experiments with human subjects. HRR has also been successfully applied to representing word order and meaning in a unified way, based on exposure to language data (Jones & Mewhort 2007).

CONCLUSION

This article has presented two popular methods for representing structured information in a distributed way: Recursive Auto-Associative Memory (RAAM) and Vector Symbolic Architectures (VSA). The main difference between the two approaches lies in the way that representations are learned: RAAM uses an iterative algorithm to converge on a single stable representation for a set of structures, whereas VSA assembles structures in a "one-shot" learning stage, but the representation of each such structure must be stored separately. While no single approach is likely to solve all or even most of the challenges of AI, there is little doubt that the advantages provided by distributed representations – robustness to noise, fast comparison without decomposition, and other psychologically realistic features – will continue to make them attractive to anyone interested in how the mind works.

REFERENCES

Ackley, D. H, Hinton, G., & Sejnowski, T. (1985) A Learning Algorithm For Boltzmann Machines. *Cognitive Science* 9, 147–169.

Anderson, J., Silverstein, J., Ritz, S., & Jones, R. (1977) Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model. *Psychological Review* 84 (5), 413–451.

Barnsley, M. F. (1993) *Fractals Everywhere*. New York: Academic Press.

Bellman, R. (1961) *Adaptive Control Processes*. Princeton University Press.

Callan, R. E. (1996) *Netting the Symbol: Analytically Deriving Recursive Connectionist Representations of Symbol Structures*. Ph.D. Dissertation, Systems Engineering Faculty, Southampton Institute, England.

Chalmers, D. (1990). Syntactic Transformations On Distributed Representations. *Connection Science* 2, 53–62.

Dominey P.F., Hoen, M., Inui, T. (2006) A Neurolinguistic Model of Grammatical Construction Processing. *Journal of Cognitive Neuroscience* 18, 2088-2107.

Eliasmith, C. & Thagard, P. (2001) Integrating Structure and Meaning: A Distributed Model of Analogical Mapping. *Cognitive Science* 25(2), 245–286.

Elman, J. (1990) Finding Structure in Time. *Cognitive Science* 14, 179–211.

Fodor, J. & Pylyshyn, Z. (1988) Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition* 28, 3–71.

Gayler, R. (2003) Vector Symbolic Architectures Answer Jackendoff's Challenges for Cognitive Neuroscience. In Slezak, P., ed.*: ICCS/ASCS International Conference on Cognitive Science. CogPrint*s, Sydney, Australia, University of New South Wales, 133–138.

Hinton, G. (1984) Distributed Representations. *Technical Report CMU-CS-84-157*, Computer Science Department, Carnegie Mellon University.

Hopfield, J. (1982) Neural Networks And Physical Systems with Emergent Collective Computational

Abilities. *Proceedings of the National Academy of Sciences* 79, 2554–2558.

Jones, M.N. & Mewhort, D.J.K. (2007) Representing Word Meaning and Order Information in a Composite Holographic Lexicon. *Psychological Review* 114 (1), 1-37.

Kanerva, P. (1994) The Binary Spatter Code for Encoding Concepts at Many Levels. In Marinaro, M. & Morasso, P., eds.: *ICANN '94: Proceedings of the International Conference on Artificial Neural Networks,* Volume 1. London: Springer-Verlag, 226–229.

Levy, S. & Pollack, J. (2001). Infinite RAAM: A Principled Connectionist Substrate for Cognitive Modeling. *Proceedings of ICCM2001*. Lawrence Erlbaum Associates.

McClelland, J., Rumelhart, D., & Hinton, G.: (1986) The Appeal of Parallel Distributed Processing. In Rumelhart, D., McClelland, J., eds.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1. MIT Press, 3-44.

Melnik, O., Levy, S. & Pollack, J.B. (2000). RAAM for Infinite Context-Free Languages. *Proceedings of IJCNN 2000*, IEEE press.

Plate, T. (1991) Holographic Reduced Representations. Technical Report CRG-TR-91-1, Department of Computer Science, University of Toronto.

Rumelhart, D.E., Hinton, G., & Williams, R.J. (1986) Learning Internal Representations by Error Propagation. . In Rumelhart, D., McClelland, J., eds.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Volume 1. MIT Press, 318-362.

Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence 36*, 77–105.

Smolensky, P. (1990) Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems. *Artificial Intelligence* 46, 159–216.

Tomasello, M. (1992). First Verbs: A Case Study In Early Grammatical Development. Cambridge University Press.

vanGelder, T. (1999) Distributed Versus Local Representation. In: *The MIT Encyclopedia of Cognitive Sciences*. MIT Press, 236–238.

Voegtlin, T. & Dominey, P.F. (2005) Linear Recursive Distributed Representations. *Neural Networks* 18, 878-895.

D

KEY TERMS

Binary Spatter Codes (BSC): VSA using bit vectors and element-wise exclusive-or (XOR) or multiplication for role/filler binding.

Binding: In its most general sense, a term used to describe the association of values with variables. In AI and cognitive science, the variables are usually a closed set of roles (*AGENT, PATIENT, INSTRUMENT*) and the values an open set of fillers (entities).

Bundling: VSA operation for combining several items into a single item, through vector addition.

Cleanup Memory: Mechanism required to compensate for noise introduced by lossy compression in VSA.

Distributed Representation: A general method of representing and storing information in which the representation of each item is spread across the entire memory, each memory element simultaneously stores the components of more than one item, and items are retrieved by their content rather than their address.

Holographic Reduced Representation (HRR): The most popular variety of VSA, uses circular convolution to bind fillers to roles and circular correlation to recover the fillers or roles from the bindings.

Recursive Auto-Associative Memory (RAAM): Neural network architecture that uses vector/matrix multiplication for binding and iterative learning to encode structures.

Tensor Products: Early form of VSA that uses the outer (tensor) product as the binding operation, thereby increasing the dimensionality of the representations without bound.

Vector Symbolic Architecture (VSA): General term for representations that use large vectors of random numbers for roles and fillers, and fast, lossy compression operations to bind fillers to roles.

EA Multi-Model Selection for SVM

Gilles Lebrun
University of Caen Basse-Normandie, France

Olivier Lezoray
University of Caen Basse-Normandie, France

Christophe Charrier
University of Caen Basse-Normandie, France

Hubert Cardot
University François-Rabelais of Tours, France

INTRODUCTION

Evolutionary algorithms (**EA**) (Rechenberg, 1965) belong to a family of stochastic search algorithms inspired by natural evolution. In the last years, EA were used successfully to produce efficient solutions for a great number of hard optimization problems (Beasley, 1997). These algorithms operate on a population of potential solutions and apply a survival principle according to a **fitness** measure associated to each solution to produce better approximations of the optimal solution. At each iteration, a new set of solutions is created by selecting individuals according to their level of fitness and by applying to them several operators. These operators model natural processes, such as selection, recombination, mutation, migration, locality and neighborhood. Although the basic idea of **EA** is straightforward, solutions coding, size of population, fitness function and operators must be defined in compliance with the kind of problem to optimize.

Multi-class problems with binary **SVM** (Support Vector Machine) classifiers are commonly treated as a decomposition in several binary sub-problems. An open question is how to properly choose all models for these sub-problems in order to have the lowest error rate for a specific SVM multi-class scheme. In this paper, we propose a new approach to optimize the **generalization capacity** of such SVM multi-class schemes. This approach consists in a global selection of models for sub-problems altogether and is denoted as **multi-model selection**. A multi-model selection can outperform the classical individual model selection used until now in the literature, but this type of selection defines a hard optimisation problem, because it corresponds to a search

a efficient solution into a huge space. Therefore, we propose an adapted **EA** to achieve **that multi-model selection** by defining specific **fitness** function and **recombination operator**.

BACKGROUND

The multi-class classification problem refers to assigning a class to a feature vector in a set of possible ones. Among all the possible inducers, Support Vector Machine (SVM) have particular high generalization abilities (Vapnik, 1998) and have become very popular in the last few years. However, **SVM** are binary classifiers and several **combination schemes** were developed to extend **SVM** for problems with more two classes (Rifkin & Klautau, 2005). These schemes are based on different principles: probabilities (Price, Knerr, Personnaz & Dreyfus, 1994), error correcting codes (Dietterich, & Bakiri, 1995), correcting classifiers (Moreira, & Mayoraz, 1998) and evidence theory (Quost, Denoeux & Masson, 2006). All these **combination schemes** involve the following three steps: 1) decomposition of a multi-class problem into several binary sub-problems, 2) **SVM** training on all sub-problems to produce the corresponding binary decision functions and 3) **decoding strategy** to take a final decision from all binary decisions. Difficulties rely on the choice of the **combination scheme** (Duan & Keerthi, 2005) and how to optimize it (Lebrun, Charrier, Lezoray & Cardot, 2005).

In this paper, we focus on step 2) when steps 1) and 3) are fixed. For that step, each binary problem needs to properly tune the SVM hyper-parameters (model) in

order to have a global low multi-class error rate with the combination of all binary decision functions involved in. The search for efficient values of hyper-parameters is commonly designed by the term of *model selection*. The classical way to achieve optimization of multi-class schemes is an individual model selection for each related binary sub-problem. This methodology overtones that a multi-class scheme based on SVM combination is optimal when each binary classifier involved in that **combination scheme** is optimal on the dedicated binary problem. But, if it is supposed that a **decoding strategy** can more or less easily correct binary classifiers errors, then individual binary model selection on each binary sub-problem cannot take into account error correcting possibilities. For this main reason, we are thinking that another way to achieve optimization of multi-class schemes is a global **multi-model selection** for binary problems altogether. In fact, the goal is to have a minimum of errors on a muti-class problem. The selection of all sub-problem models (multi-model selection) has to be globally performed to achieve that goal, even if that means that error rates are not optimal on all binary sub-problems when they are observed individually. **EA** is an efficient meta-heuristic approach to realize that **multi-model selection**.

EA MULTI-MODEL SELECTION

This section is decomposed in 3 subsections. In the first section, the multi-model optimization problem for muticlass **combination schemes** is exposed. More details than in previous section and useful notations for next subsections are introduced. In the second section, our EA multi-model selection is exposed. Details on fitness estimation of multi-model and crossover operator over them are described. In the third section, experimental protocol and results with our EA multi-model selection are provided.

Multi-Model Optimization Problem

A multi-class combination scheme induces several binary sub-problems. The number k and the nature of binary sub-problems depend on the decomposition involved in the combination scheme. For each binary sub-problem, a **SVM** must be trained to produce an appropriate binary decision function h_i ($1 < i < k$). The quality of h_i is greatly dependent on the selected model

θ_i and is characterized by the expected error rate e_i for new datasets with the same binary decomposition. Each model θ_i contains all hyper-parameters values for training a SVM on dedicated binary sub-problem. Expected error rate e_i associated to a model θ_i is commonly determined by cross-validation techniques. All the θ_i models constitute the multi-model $\theta = (\theta_1,..., \theta_k)$. The expected error rate e of a SVM multi-class combination scheme is directly dependent on the selected multi-model θ. Let Θ denote the multi-model space for a multi-class problem (*i.e.* $\forall \theta : \theta \in \Theta$) and Θ_i the model space for the ith binary sub-problem. The best θ^* multi-model is the one for which expected error e is minimum and corresponds to the following optimization problem:

$$\theta^* = \arg \min_{\theta \in \Theta} e(\theta) \qquad (1)$$

where $e(\theta)$ denotes the expected error e of a multi-class combination scheme with the multi-model θ. The huge size of the multi-model space ($\Theta = \times_{i \in [1,k]} \Theta_i$) makes the optimization problem (0.1) very hard. To reduce the optimization problem complexity, it is classic to use the following approximation:

$$\tilde{\theta} = \{\arg \min_{\theta \in \Theta} e(\theta_i) \mid i \in [1, k]\} \qquad (2)$$

Hypothesis is made that

$$e(\tilde{\theta}) \approx e(\theta^*).$$

This hypothesis also supposes that

$$e(\tilde{\theta}_i) \approx e(\theta_i^*).$$

If it is evident that each individual model θ_i in the best multi-model θ^* must correspond to efficient **SVM** (*i.e.* low value of e_i) on the corresponding i^{th} binary subproblem, all best individual models ($\theta_1^*,...,\theta_k^*$) do not necessarily define the best multi-model θ^*. The first reason is that all error rates e_i are estimated with some tolerance and combination of all these deviations can have a great impact on the final multi-class error rate e. The second reason is that even if all the binary classifiers of a **combination scheme** have identical e_i error rates for different multi-models, these binary classifiers can have different binary class predictions for a same

example according to the used multi-model. Indeed multi-class predictions by combining these binary classifiers could be different for a same feature vector example since the correction involved in a given **decoding strategy** depends on the nature of the internal errors of the binary classifiers (mainly, the number of errors). Then, multi-class classification schemes with the same internal-errors e_i, but different multi-models θ, can have different capacities of generalization. For all these reasons, we claim that multi-model optimization problem (0.1) can outperform individual model optimization (0.2).

Evolutionary Optimization Method

Within our AE multi-model selection method, a **fitness** measure f is associated to a multi-model θ which is all the more large as the error e associated to θ is small; this enables to solve (0.1) optimization problem. **Fitness** value is normalized in order to have $f=1$ when error e is zero and $f=0$ when error e corresponds to a random draw. Moreover, the number of examples in each class are not always well balanced for many multi-class datasets; to overcome this, the error e corresponds to a Balanced Error Rate (**BER**). As regards these two key points, the proposed fitness formulation is:

$$f = \frac{1}{1 - \frac{1}{n_c}} \left(1 - \frac{1}{n_c} - e \right) \tag{3}$$

with n_c denoting the number of classes in a multi-class problem. In the same way, the internal-fitness f_i is defined as $f_i = 1 - 2e_i$ for the ith binary classifier with corresponding **BER** e_i.

The EA **crossover operator** for the combination of two multi-models θ¹ and θ² must favor the selection of most efficient models in these two multi-models. It is worth noting that one should not systematically select all the best models to produce an efficiency child multi-model θ as explained in previous sub-section. For each sub-problem, internal-fitness f_i^1 and f_i^2 are used to determine the probability

$$p_i = \frac{\left(f_i^1 \right)^2}{\left(f_i^1 \right)^2 + \left(f_i^2 \right)^2} \tag{4}$$

to select the ith model in θ¹ as the ith model in the child multi-model θ. f_i^j denotes the internal fitness of the ith binary classifier with the multi-model $θ^j$. For the child multi-models generated by the **crossover operator**, an important advantage is that no new SVM training is necessary if all the related binary classifiers were already trained. In contrast, only the **BER** error rates of all child multi-models have to be evaluated. **SVM** Training is only necessary for the first step of the **EA** and when models go through a mutation operator.

The rest of our **EA** for **multi-model selection** is similar to other **EA** approaches. First, at initialization step, a population of λ multi-models is generated at random. Each model $θ_i^j (1 \leq i \leq λ, 1 \leq j \leq k)$ corresponds to an uniform random within all possible values of SVM hyper-parameters. New multi-models are produced by combination of multi-models couples selected by a Stochastic Universal Sampling (**SUS**) strategy. A fixed selective pressure p_s is used for the **SUS** selection. Each model $θ_i^j$ has a probability of p_m/k to mutate (uniform random as for the initialization step of EA). **Fitness** f of all child multi-models are then evaluated. A second selection step is used to define the population of the next iteration of our **EA**. λ individuals are selected by a **SUS** strategy (same selective pressure p_s is used) from both the λ parents and the λ children. Its become the multi-models population in the next iteration. The number of iterations of **EA** is fixed to n_{max}. At the end of the **EA**, the multi-model with the best **fitness** f from all these iterations is selected as $θ^*$.

Experimental Results

In this section, three well known multi-class datasets are used: Satimage ($n_c = 6$), Letter ($n_c = 26$) from the Statlog collection (Blacke & Merz, 1998), and USPS ($n_c = 10$) dataset (Vapnik, 1998). In (Wu, Lin & Weng, 2004), two sampling sizes of 300/500 and 800/1000 are used to constitute training/testing datasets. For each sampling sizes, 20 random splits are generated. We have used the same sampling sizes and the same split for the 3 datasets: Satimage, Letter and USPS. Two optimization methods are used for the selection of the best multi-model $θ^*$ for each training datasets. The first one is the classical individual model selection and the second one is our **EA multi-model selection**. For both methods, two combination schemes are used:

Table 1. Average BER with individual model selection (column $\bar{e}_{classic}$) and our EA multi-model selection (column $\bar{e}_{EA}$). Negative values in column $\Delta\bar{e}(\Delta\bar{e} = \bar{e}_{EA} - \bar{e}_{classic})$ correspond to an improvement of the performance of a multi-class combination scheme when our EA multi-model selection method is used.

E

Size	500			1000		
	$\bar{e}_{classic}$	$\bar{e}_{EA}$	$\Delta\bar{e}$	$\bar{e}_{classic}$	$\bar{e}_{EA}$	$\Delta\bar{e}$
one-versus-one						
Satimage	14.7 ± 1.8 %	14.5 ± 2.1 %	-0.2 %	11.8 ± 0.9 %	11.8 ± 1.0 %	-0.0 %
USPS	12.8 ± 1.2 %	11.0 ± 1.8 %	-1.8 %	8.9 ± 0.9 %	8.4 ± 1.6 %	-0.5 %
Letter	40.5 ± 3.0 %	35.9 ± 2.9 %	-4.6 %	21.4 ± 1.7 %	18.6 ± 2.1 %	- 2.8 %
one-versus-all						
Satimage	14.6 ± 1.7 %	14.5 ± 2.0 %	-0.1 %	11.5 ± 0.8 %	11.6 ± 1.0 %	+0.1 %
USPS	11.9 ± 1.3 %	11.2 ± 1.5 %	-0.7 %	8.8 ± 1.3 %	8.5 ± 1.6 %	-0.3 %
Letter	41.9 ± 3.3 %	36.3 ± 3.3 %	-5.6 %	22.1 ± 1.3 %	19.7 ± 1.8 %	-2.4 %

one-versus-one and *one-versus-all* (Rifkin & Klautau, 2004)[1]. For each binary problem, a SVM with Gaussian kernel $K(u,v) = \exp(-\gamma\|u - v\|^2)$ is trained (Vapnik, 1998). Possible values of SVM hyper-parameters for a model are C trade-off SVM constant (Vapnik, 1998) and widthband γ of gaussian kernel function ($\theta_i \equiv (C_i, \gamma_i)$). For all binary problems: $\theta_i \in \Theta_i = [2^{-5}, 2^{-3},...,2^{15}] \times [2^{-5}, 2^{-3},...,2^{15}]$. Individual space model Θ_i is based on grid search techniques (Chang & Lin, 2001). **BER** e on a multi-class problem and **BER** e_i on binary sub-problems are estimated by five-fold **cross-validation** (CV). These **BER** values are used by our **EA** for the **multi-model selection**. Final **BER** e of a selected multi-model by our **EA** is estimated on a test datasets not used during the **multi-model selection** process. Our EA has several constants that must be fixed and we have made the following choices: $p_s = 2$, $\lambda = 50$, $n_{max} = 100$, $p_m = 0.01$.

Table 1 gives average **BER** under all 20 split sets of previously mentioned datasets for each training set size (row size of table 1). This is done for the two combination schemes (*one-versus-one* and *one-versus-all*), and for the two above mentioned selection methods (columns $\bar{e}_{classic}$ and $\bar{e}_{EA}$). Column $\Delta\bar{e}$ provides the average variation of **BER** between our **multi-model selection** and classical one. Results of that column are particularly important. For two datasets (USPS and Letter) our

optimization method produces SVM **combination schemes** with best **generalization capacities** than the classical one. That effect appears to be more marked when number of classes in the multi-class problem increases. A reason is that the multi-model space search size exponentially increases with the number k of binary problems involved in a **combination scheme** (121^k for those experiments). This effect is directly linked to the number of classes n_c and could explain why improvements are not measurable with Satimage dataset. In some way, a classical optimization method explores the multi-model space Θ in blink mode, because cumulate effect of the combination of k **SVM** decision functions could not be determined without estimation of e. That effect is emphasized when estimated **BER** e_i are poor (*i.e.* training and testing data size are low). Comparison of $\Delta\bar{e}$ values when training/testing dataset size change in table 1 illustrates this one.

FUTURE TRENDS

The proposed **EA multi-model selection** method has to be tested with other **combination schemes** (Rifkin & Klautau, 2004), like error-correcting output codes in order to measure their influence. Effect with others datasets, which have a great range in number of classes,

must also be tested. Adding feature selection (Fröhlich, Chapelle & Schölkopf, 2004) abilities to our AE muti-model selection is also of importance.

Another key point to take into account is the reduction of the learning time of our **EA** method which is actually expensive. One way to explore this is to use fast **CV** error estimation technique (Lebrun, Charrier, Lezoray & Cardot, 2006) for the estimation of BER.

CONCLUSION

In this paper, a new **EA multi-model selection** method is proposed to optimize the generalization capacities of SVM **combination schemes**. The definition of a **cross-over** operator based on internal **fitness** of **SVM** on each binary problem is the core of our **EA** method. Experimental results show that our method increases the **generalization capacities** of *one-versus-one* and *one-versus-all* **combination schemes** when compared with individual model selection method.

REFERENCES

Beasley, D. (1997). *Possible applications of evolutionary computation.* Handbook of Evolutionary Computation. 97/1, A1.2. IOP Publishing Ltd. And Oxford University Press.

Blacke, C., & Merz, C., (1998). *UCI repository of machine learning databases. Advances in Kernel Methods, Support Vector Learning.* University of California, Irvine, Dept. of Information and Computer Sciences.

Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines.* Software available at http://www.csie.ntu.edu.tw/\~cjlin/libsvm.

Dietterich, T. G., & Bakiri, G. (1995). *Solving Multiclass Learning Problems via Error-Correcting Output Codes.* Journal of AI Research. (2) 263-286.

Duan, K.-B., & Keerthi, S. S. (2005). *Which Is the Best Multiclass SVM Method? An Empirical Study.* Multiple Classifier Systems. 278-285.

Fröhlich, F., Chapelle, O., & Schölkopf, B. (2004). Feature Selection for Support Vector Machines Using Genetic Algorithms. International Journal on Artificial Intelligence Tools. 13(4) 791-800.

Lebrun, G., Charrier, C., Lezoray, O., & Cardot, H. (2005). *Fast Pixel Classification by SVM Using Vector Quantization, Tabu Search and Hybrid Color Space.* Computer Analysis of Images and Patterns. (LNCS, Vol. 3691) 685-692.

Lebrun, G., Charrier, C., Lezoray, O., & Cardot, H. (2006). *Speed-up LOO CV with SVM classifier.* Intelligence Data Engineering and Automated Learning. (LNCS, Vol. 4224) 108-115.

Lebrun, G., Charrier, C., Lezoray, O., & Cardot, H. (2007). *An EA multi-model selection for SVM multiclass schemes.* Computational and Ambient Intelligence. 260-267 (LNCS, Vol. 4507).

Moreira, M., & Mayoraz, E. (1998). *Improved Pairwise Coupling Classification with Correcting Classifiers.* European Conference on Machine Learning. 160-171.

Price, D., Knerr, S., Personnaz, L., & Dreyfus, G. (1994). *Pairwise Neural Network Classifiers with Probabilistic Outputs.* Neural Information Processing Systems. 1109-1116.

Quost, B., Denoeux, T., & Masson, M. (2006). *One-against-all classifier combination in the framework of belief functions.* Information Processing and Management of Uncertainty in Knowledge-Based Systems. (1) 356-363.

Rechenberg, I. (1965). *Cybernetic Solution Path of an Experimental Problem.* Royal Aircraft Establishment Library Translation.

Rifkin, R., & Klautau, A. (2004). *In Defense of One-Vs-All Classification.* Journal of Machine Learning Research. (5) 101-141.

Vapnik, V.N. (1998). *Statistical Learning Theory.* Wiley Edition.

Wu, T.-F., Lin, C.-J., & Weng, R. C., (2004). *Probability Estimates for Multi-class Classification by Pairwise Coupling.* Journal of Machine Learning Research. (5) 975-1005.

KEY TERMS

Cross-Validation: A method of estimating predictive error of inducers. Cross-validation procedure splits

that dataset into *k* equal-sized pieces called folds. *k* predictive function are built, each tested on a distinct fold after being trained on the remaining folds.

Evolutionary Algorithm (EA): Meta-heuristic optimization approach inspired by natural evolution, which begins with potential solution models, then iteratively applies algorithms to find the fittest models from the set to serve as inputs to the next iteration, ultimately leading to a sub-optimal solution which is close to the optimal one.

Model Selection: Model Selection for Support Vector Machines concerns the tuning of SVM hyper-parameters as *C* trade-off constant and the kernel parameters.

Multi-Class Combination Scheme: A combination of several binary classifiers to solve a given multiclas problem.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in.

Support Vector Machine (SVM): SVM maps input data in a higher dimensional feature space by using a non linear function and finds in that feature space the optimal separating hyperplane maximizing the margin (that is the distance between the hyperplane and the closest points of the training set) and minimizing the number of misclassified patterns.

Trade-Off Constant of SVM: The trade-off constant, noted *C*, permit to fix the importance to increase the margin for the selection of optimal hyper-plan in comparison with reducing predictive errors (i.e. examples which not respect margin distance from hyper-plan separator).

ENDNOTE

[1] More details on used combinations schemes are given in (Lebrun, Lezoray, Charrier & Cardot, 2007).

EC Techniques in the Structural Concrete Field

Juan L. Pérez
University of A Coroña, Spain

Belén González-Fonteboa
University of A Coroña, Spain

Fernando Martínez-Abella
University of A Coroña, Spain

INTRODUCTION

Throughout the last decades, one of society's concerns has been the development of new tools to optimize every aspect of daily life. One of the mechanisms that can be applied to this effect is what is nowadays called Artificial Intelligence (AI). This branch of science enables the design of intelligent systems, meaning that they display features that can be associated to human intelligence, search methods being one of the most remarkable. Amongst these, Evolutionary Computation (EC) stands out. This technique is based on the modelling of certain traits of nature, especially the capacity shown by living beings to adapt to their environment, using as a starting point Darwin's Theory of Evolution following the principle of natural selection (Darwin, 1859). These models search for solutions in an automatized way. As a result, a series of search techniques which solve problems in an automatized and parallel way has arisen. The most successful amongst these are Genetic Algorithms (GA) and, more recently, Genetic Programming (GP). The main difference between them is rooted on the way solutions are coded, which implies certain changes in their processing, even though the operation in both systems is similar.

Like most disciplines, the field of Civil Engineering is no stranger to optimization methods, which are applied especially to construction, maintenance or rehabilitation processes (Arciszewski and De Jong, 2001) (Shaw, Miles and Gray, 2003) (Kicinger, Arciszewski and De Jong, 2005). For instance, in Structural Engineering in general and in Structural Concrete in particular, there are a number of problems which are solved simultaneously through theoretical studies, based on physical models, and experimental bench-marks which sanction and adjust the former, where a large amount of factors intervene. In these cases, techniques based on Evolutionary Computation are capable of optimizing constructive processes while accounting for structural safety levels. In this way, for each particular case, the type of materials, their amount, their usage, etc. can be determined, leading to an optimal development of the structure and thus minimizing manufacturing costs (Rabuñal , Varela, Dorado, González and Martínez, 2005).

GENETIC ALGORITHMS

At the origin of what is now known as Genetic Algorithms are the works of John Holland at the end of the 1960's. He initially named them "Reproductive Genetic Planning", and it wasn't until the 70's that they received the name under which they are known today (Holland, 1975).

GA is a search algorithm inspired on the biological functioning of living beings. It is based upon reproductive processes and the principle which determines that better environmetally adapted individuals have more chances of surviving (Goldberg, 1989).

Like living beings, GAs use the basic heritage unit, the gene, to obtain a solution to a problem. The full set of genes (parameters characterizing the problem) is chromosome, and the expression of the chromosome is an individual in particular.

In Computer Science terms, the representation of each individual is a chain, usually binary, assigning a certain number of bits to each parameter. For each variable represented a conversion to discrete valued has to be performed. Obviously, not all parameters have

to be coded with the same number of bits. Each one of the bits in a gene is usually called allele. Once the individuals' genotype (the structure for the creation of an individual in particular) is defined, we are ready to carry out the evolutionary process that will reach the solution to the problem posed.

We start with a random set of individuals, called a population. Each of these individuals is a potential solution to the problem. This would be the initial population or zero generation, and successive generations will be created from it until the solution is reached. The mechanisms used in the individuals' evolution are analogous to the functioning of living beings:

- **Selection** of individuals for reproduction. All selection algorithms are based on the choice of individuals by giving higher survival probabilities to those which offer a better solution to the problem, but allowing worse individuals to be also selected so genetic diversity will not be lost. Unselected individuals will be copied through to the following generation.
- Once the individuals have been selected, **crossover** is performed. Typically, two individuals (parents) are crossed to produce two new ones. A position is established before which the bits will correspond to one parent, with the rest belonging to the other. This crossover is named single point crossover, but a number of points could be used, where bit subchains separated by points would belong alternatively to one or the other parent.
- Once the new individuals have been obtained, small variations in a low percentage of them are performed. This is called **mutation**, and its goal is to carry out an exploration in the state space.

Once the process is over, the new individuals will be inserted in the new population, constituting the next generation.

New generations will be produced until the population has created a sufficiently adequate solution, the maximum number of generations has been reached, or the population has converged and all individuals are equal.

GENETIC PROGRAMMING

Genetic Programming (GP), like GAs, is a search mechanism inspired on the functioning of living beings. The greater difference between both methods consists in the way solutions are coded. In this case, it is carried out as a tree structure (Koza, 1990) (Koza, 1992). The main goal of GP is to produce solutions to problems through program and algorithm induction.

The general functioning is similar to that of the GAs. Nevertheless, due to the difference in solution coding, great variations exist in the genetic operations of initial solution generation, crossover and mutation. The rest of operations, selection and replacement algorithms, remain the same, as do the metrics used to evaluate individuals (fitness).

We will now describe two cases where both techniques have been applied. They refer to questions related to Structural Concrete, approached to as both a material and a structure.

EXAMPLE 1: Procedure to determine optimal mixture proportion in High Performance Concrete.

High Performance Concrete (HPC) is a type of concrete designed to attain greater durability together with high resistance and good properties in its fresh state to allow ease of mixing, placing and curing (Forster, 1994) (Neville and Aitcin, 1998) . Its basic components are the same of ordinary concrete, with the intervention in diverse quantities of additions (fly ash or silica fume, byproducts of other industries that display pozzolanic resistance capacities) plus air-entraining and/or fluidifying admixtures. Indeed, an adequate proportion of these components with quality cement, water and aggregates produces concrete with excellent behavior, generally after an experimental process to adjust the optimal content in each material. When very high resistance is not a requirement, the addition introduced is fly ash (FA) ; air-entraining admixtures (AE) are used to improve behavior in frost/defrost situations. When high resistance is needed, FA is substituted by a silica fume (SF) addition, eliminating AE altogether. In every case high cement contents and low water/binder (W/B) ratios are used (both cement and pozzolanic additions are considered binders).

A number of mixture proportioning methods exist, based on experimental approaches and developed by different authors. The product of such mixtures can

be controlled through various tests, two of which are particularly representative of the fresh and hardened states of concrete: the measurement of workability and evaluation of compressive strength, respectively. The first one is carried out through the slump test, where the subsidence of a molded specimen after removal of the mold is measured. A high value ensures adequate placing of concrete. The second test consists on the compression of hardened concrete until failure is produced; HPC can resist compressive stresses in the range of 40 to 120 MPa.

The goal of mixture proportioning methods is to adjust the amount of each component to obtain slump and strength values within a chosen range. There is a large body of experience, and basic mixtures which require some experimental adjustment are available. It is difficult to develop a theoretical model to predict slump and resistance of a particular specimen, though roughly adequate fitted curves do exist (Neville, 1981). It is even harder to approach theoretically the reverse problem, that is, to unveil mixture proportioning starting from the goal results (slump and strength).

Chul-Hyun Lim et al. (Chul-Hyun, Young-Soo and Joong-Hoon, 2004) have developed a GA based application to determine the relationship between different parameters used to characterize a HPC mixture. Two types of mixture are considered regarding their goal strength: mixtures that reach values between 40 and 80 MPa and mixtures that reach between 80 and 120 MPa. If a good database is available, it is not hard to obtain correlations between different variables to predict slump and strength values for a particular specimen. Nevertheless, the authors use GAs to solve the reverse problem, that is, to obtain mixture parameters when input data are slump and strength.

To that effect they use a database with 104 mixtures in the 40 to 80 MPa range and 77 in the 80 to 120 MPa range. In the first group, essential parameters are, according to prior knowledge, W/B ratio (%), amount of water (W, kg/m^3), fine aggregate to total aggregate ratio (s/a, %), amount of AE admixture (kg/m^3), amount of cement replaced by FA (%), and amount of high-range water-reducing admixture (superplasticizer, SP, kg/m^3). Tests on the different mixtures give slump values (between 0 and 300 mm) and compressive strength. In the second group, essential parameters are W/B, W, s/a, amount of cement replaced by silica fume (SF, %) and SP. Tests give out slump and resistance values.

Using multiple regression techniques, the authors firstly obtain for each group two fitted curves group that predict slump and strength from starting variables. GAs are used to solve the reverse problem. For each group, they first reach an individual which determines optimal W/B, W, s/a, FA and AE, or W/B, W, s/a and SF for a specific compressive strength. From these parameters, using the prediction curve previously obtained, the optimal SP value for a particular slump is calculated.

The development of genetic algorithms used is based on programs by Houck et al. (Houck, Joines, and Kay 1996) In this case, "ranking selection based on normalized geometric distribution" has been used for individual selection. One-point, two-point, and uniform crossover algorithms are used. Different strategies are used for mutation operations: boundary mutation, multi-nonuniform mutation, nonuniform mutation, and uniform mutation. The first trials were carried out with an initial population consisting of only 15 individuals, which lead to local minima only. Optimal results are obtained increasing population to 75 individuals.

It should be pointed out that the reverse problem is thus solved in two phases. In the first phase all component amounts are fixed except for SP, with compressive strength as a target; following this, SP is fixed to attain the desired slump. Initial fitting functions are used as a simple but accurate approach to the real database, to avoid using it directly, which would make solving the reverse problem a more difficult task.

For the first group, highest errors correspond to AE and SP determination, up to 12.5% and 15% respectively. Errors committed in the second group are smaller. In any case, errors are relatively minor since these materials are included as admixtures in very small amounts. As a conclusion for this example, it is interesting to point out that the procedure is not only a useful application of GAs to concrete mixture proportioning, but also constitutes by itself a new mixture proportioning method that requires GAs for its application.

EXAMPLE 2: Determination of shear strength in reinforced concrete deep beams

Deep beams are those that can be considered short (span, L) in relation to their depth (h). There is no consensus in international codes as to what is the span-to-depth threshold value dividing conventional and deep beams. In the example shown here, L/h ratios between 0.9 and

Figure 1. Deep beam parameters

E
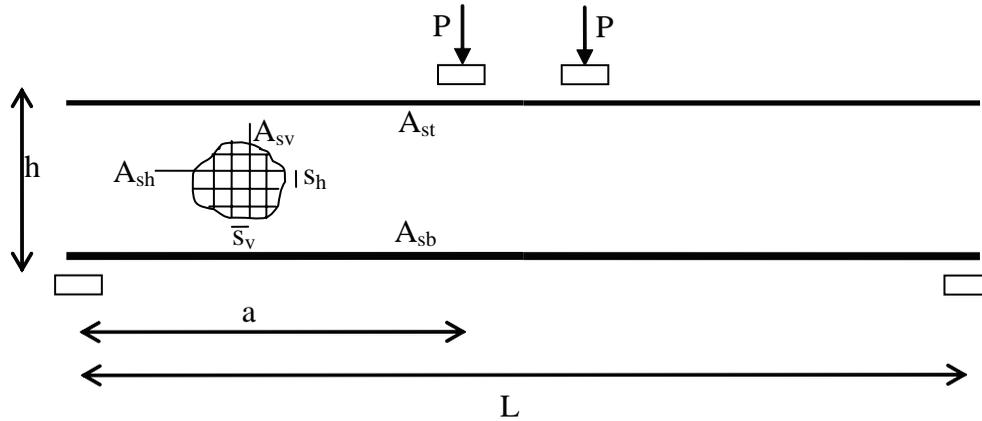
4.14 are considered for simply supported elements working under two point loads as shown in Figure 1. Shear failure is a beam collapse mode produced by the generation of excesssive shear stresses in the vicinity of the supports. Its value depends on various parameters commonly used for beam design (see Figure 1): a/h, L/h, section area (A_{sv} and A_{sh}), strength (f_{yv} and f_{yh}) and distances between vertical and horizontal steel rebars (s_v y s_h) placed in the failure zone, which will be respectively called vertical and horizontal transverse reinforcement; section area (A_{sb} and A_{st}) and strength (f_{yb} and f_{yt}) of longitudinal rebars placed in the lower and higher zones of the beam, which will be respectively called bottom and top reinforcement; compressive strength of concrete used in the beam, and width (b) of the latter.

The dependence of shear strength on these parameters is known through multiple studies (Leonhardt and Walther, 1970) and it can be presented in a normalized form through simple relationships. Relevant parameters are reduced to the following:

- $X_1 = a/h$
- $X_2 = L/h$
- $X_3 = (A_{sv}f_{yv})/(bs_vf_c)$
- $X_4 = (A_{sh}f_{yh})/(bs_hf_c)$
- $X_5 = (A_{sb}f_{yb})/(bhf_c)$
- $X_6 = (A_{st}f_{yt})/(bhf_c)$

Normalized shear strength can be written as $R = P/(bhf_c)$, where P is failure load (Figure 1).

Ashour et al. (Ashour, Alvarez and Toropov, 2003) undertake the task of finding an expression that can predict shear strength from these variables. It is not easy to develop an accurate mathematical model capable of predicting shear strength. Usual curve fitting techniques do not produce good results either, though they have been the base of diverse international codes that include design prescriptions for these structural elements. GP appears to be a tool that can reach consistent results. The authors develop various expressions with different complexity levels, obtaining different degrees of accuracy. A database of 141 tests available in scientific literature is used.

A remarkable feature of GP techniques is that if the possibility of using complex operators is introduced, fitting capacity is impressive when variables are well distributed and their range is wide. Notwithstanding, if the goal of the process is to obtain an expression that can be used by engineers, one of its requirements is simplicity. The authors of this study take into account this premise and only choose as operators addition, multiplication, division and squaring. A first application leads to a complex expression that reveals the very low influence of parameter X_2, which is thus eliminated. With the 5 remaining variables, a simple and accurate expression is obtained (root mean square –RMS- training error equal to 0.033; average ratio between predicted and real R equal to 1.008 and standard deviation equal

to 0.23). Validation of this expression is performed whit 15 new tests and also found to be accurate: RMS error equal to 0.035; average ratio between predicted and real R values equal to 1.11 and standard deviation equal to 0.21.

As a conclusion to this example, it can be pointed out that expressions obtained through GP bring as an added value that they can become virtual laboratories. Indeed, by fixing one or multiple variables, the influence on the response of the variation of a specific one, and also determine which variables are most important in the studied phenomenon (in this case, X_1 and X_5). Finally, GP techniques prove to be a powerful tool for the development and improvement of codes that regulate concrete structure design. Even when the precise physical mechanism is unknown, GP allows for the development of accurate expressions that can be factored a posteriori to reach safety levels associated to an acceptable failure probability.

CONCLUSION

Evolutionary Computation is a valid technique for optimization and regression problems in the fields of Structural Engineering in general and Structural Concrete in particular.

In the first of the two examples analyzed, GAs have been used to determine optimal mixture proportion for High Performance Concrete, using as target data its compressive stregth and workability. In this case, evolutionary techniques show their power by solving the reverse problem, producing a new mixture proportioning method for concrete.

In the second example, GP techniques were used to accurately predict structural response of concrete beams from benchmark experimental data series. The advantage brought forth by Evolutionary Computation is the capacity to analyze physically complex phenomena, by creating a "virtual laboratory". The line of work opened towards the improvement of design codes and rules sometimes purely based on testing is also of great importance.

Application of EC techniques is growing exponentially in this field thanks to fruitful collaboration between EC and Structural Engineering experts.

REFERENCES

Arciszewski, T., and De Jong, K. A. (2001). Evolutionary computation in civil engineering: research frontiers. In B. H. V. Topping (Ed.), Proceedings of the Eight International Conference on Civil and Structural Engineering Computing, Eisenstadt, Vienna, Austria.

Ashour, A.F. Alvarez, L.F. and Toropov V.V. (2003) Empirical modelling of shear strength of RC deep beams by genetic programming. Computers and Structures 81, 331–338.

Chul-Hyun, L. Young-Soo, Y. and Joong-Hoon, K. (2004) Genetic algorithm in mix proportioning of high-performance concrete. Cement and Concrete Research 34. 409–420

Darwin, C.R. (1859) On the origin of species by means of natural selection or the preservation of favoured races in the struggle for life. Cambridge University Press, Cambridge, UK, sixth edition, 1864, originally published in 1859.

Forster, SW. (1994) High-performance concrete - Stretching the paradigm, Concrete International. 16 (10), 33–34.

Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Holland, J. H. (1975) Adaptation in Natural and Artificial Systems. Ann Arbor: The University of Michigan Press.

Houck, C.R. Joines, J. and Kay M. (1996) A genetic algorithm for function optimization: a MATLAB implementation, ACM Trans. Math. Softw.

Kicinger, R., Arciszewski, T., and De Jong, K. A. (2005). Evolutionary computation and structural design: a survey of the state of the art, Computers & Structures, 83(23-24), 1943-1978.

Koza J. (1990). Genetic Programming: A paradigm for genetically breeding populations of computer programs to solve problems. Stanford University Computer Science Department Technical Report.

Koza, J. (1992) Genetic Programming. On the Programming of Computers by means of Natural Selection. The Mit Press, Cambridge, Massachusetts, 1992.

Leonhardt, F. and Walther, R. (1970) Deep beams. Deutscher Ausschuss Für Stahlbeton Bulletin 178, Wilhelm Ernst and Sohn (Berlin), CIRIA English Translation

Neville, A.M. (1981) Properties of Concrete, 3rd ed., Pitman, London. 203– 207.

Neville, A.M. and Aitcin, P.C. (1998) High-performance concrete - An overview, Materials and Structures. 31 (3), 111 – 117.

Rabuñal, J.R. Varela, M. Dorado, J. González, B. and Martínez, I. (2005). Aplicación de la Programación Genética para determiner la adherencia en hormigón armado. Proceedings of MAEB'2005. 76-84

Shaw, D. Miles, J. C. & Gray, A. (2003). Genetic programming within civil engineering: a review. In O. Ciftcioglu & E. Dado (Eds.), Proceedings of the 10th International Workshop of the European Group for Intelligent Computing in Engineering (EG-ICE), Delft, The Netherlands, 29-39.

KEY TERMS

E

Compressive Strength: The measured maximum resistance of a concrete or mortar specimen to axial compressive loading; expressed as force per unit cross-sectional area; or the specified resistance used in design calculations.

Deep Beam: A flexural member whose span-to-depth ratio is too low to accurately apply the principles of sectional design through sectional properties and internal forces shear strength the maximum shearing stress a flexural member can support at a specific location as controlled by the combined effects of shear forces and bending moment

High Performance Concrete: Concrete meeting special combinations of performance and uniformity requirements that cannot always be achieved routinely using conventional constituents and normal mixing, placing, and curing practices.

Mixture Proportion: The proportions of ingredients that make the most economical use of available materials to produce mortar or concrete of the required properties.

Slump: A measure of consistency of freshly mixed concrete, mortar, or stucco equal to the subsidence measured to the nearest 1/4 in. (6 mm) of the molded specimen immediately after removal of the slump cone.

Superplasticizer or High-Range Water-Reducing Admixture: A water-reducing admixture capable of producing large water reduction or great flowability without causing undue set retardation or entrainment of air in mortar or concrete.

Workability: That property of freshly mixed concrete or mortar that determines the ease with which it can be mixed, placed, consolidated, and finished to a homogenous condition.

E-Learning in New Technologies

Nieves Pedreira
University of A Coruña, Spain

José Ramón Méndez Salgueiro
University of A Coruña, Spain

Manuel Martínez Carballo
University of A Coruña, Spain

INTRODUCTION

E-learning and the impact of new technologies across contemporary life is a very significant field to education. The challenge of the technology to conventional learning patterns cannot be ignored and in itself raises a host of questions: can online learning facilitate deep learning? How well does video conferencing alleviate the challenge of distance? In what ways can collaborative learning communities be developed and sustained using current and new technologies? At the same time, new communications technologies are impacting on the ways in which we understand ourselves and the worlds in which we live. Relating to this, the aim of today's education is not to learn certain contents, but rather learn to learn in the course of a whole lifetime.

The study of the learning process can help us to find the relevant points to set up some interesting characteristics of a really functional e-learning system.

THE LEARNING PROCESS

The learning process consists of a modification of our conduct that, by extracting knowledge from acquired experience, enables us to tackle problems (Pedreira, 2004a). This definition highlights the two basic aspects of all learning processes: knowledge acquisition, and the experience that leads to it.

Most studies on the nature of knowledge agree on the fact that knowledge is at the top of the hierarchical structure called information. According to this vision, data represent facts or concepts in a formalised way that allows their communication, interpretation or elaboration by human beings or by automatic means (syntactic level of the information). The so-called "news" is the meaning that an intelligent being attaches to data based on the conventional rules used for their representation (semantic level). Knowledge implies the judgement of facts and situations, and consists of inferred data and news, tacit relations between objects, concepts, events and situations, and of the necessary control actions to manage all these elements in an effective way. As such, knowledge concerns the pragmatic aspect of information because it combines the received news with the knowledge that the observer already possesses.

EDUCATION IN KNOWLEDGE SOCIETY

In recent years, so many changes have affected education that education itself needs to be updated. The amount of knowledge that we deal with is much bigger than before, the interrelations between different forms of information are much more complex, and the sources are dispersed. Such being the case, the linear model, in which each question has a place and a moment, is no longer adequate for today's information. Logical hierarchies are replaced by multiple and simultaneous media that respond to the needs of the knowledge process. The inevitable increase in complexity and quantity of the information that is available and necessary has led to a need for continuous learning.

Furthermore, in modern society, knowledge is not exclusively related to education. We live in what is called the "information or knowledge society", where the possession of knowledge is a determining factor.

Knowledge handling requires a profound transformation of learning and teaching methods: from a model in which the teacher is the monopolising agent and the authorised representative of knowledge, we must move towards a model that offers the student room for

individual exploration and self-learning. The student needs to build relations, discover the process from within, and feel stimulated to draw his own roadmap (Piaget, 1999).

This kind of learning can only be obtained through action strategies that are not perceived as restricting obligations but rather as interesting learning options. Contents, for instance, should be represented not as an object of study but rather as necessary elements towards a series of objectives that will be discovered in the course of various tests. Computer games apply the same strategy by making their users learn to proceed from one phase to another based on obtained experience and improved dexterity. This way they keep users entertained for hours in a row by trial and error.

Besides, students come from different environments and have different ages and education backgrounds, which make it more complicated to integrate them into one single group. Real personalised attention would require many more teachers and much more time. Add to that the increasing demand for continuous education, with flexible timetables and subjects, and it becomes clear that the current programmes are much too rigid.

The advantages of e-learning include convenience and portability (physical and temporal flexibility), cost and selection (wide range of courses and prices, different levels), individualisation and a higher level of student implication (WorldWideLearn, 2007).

However, if the contents of the learning platforms remain the same as those of traditional systems, even if their presentation format is adapted, they do not substantially contribute to the improvement of the learning process (Martínez, 2002). The same happens with the use of computational systems that support ex cathedra teaching and improve the acquisition of certain skills, such as simulators and games. Simulators can only be used when certain concepts are already clearly understood, and in most cases, their interface is quite complicated. Computer games are mostly used for concrete aspects and in elementary courses.

Instructional Design for e-Learning has been perfected and refined over many years using established teaching principles, with many benefits to students, but it is necessary to go on with the studies on this area because the results are still not as good as desired.

NEW TECHNOLOGIES PROPOSAL

Even so, current communication technologies, including Artificial Intelligence, allow the implementation of learning strategies based on action (e.g. videogames), the incorporation of systems that improve knowledge management (Wiig, 1995), the recuperation of the one-to-one learning model (master-apprentice becomes teacher-student), and the implementation of a new learning model ("many teachers for one student"). A computer model including all these characteristics can be a solid basis for the improvement of the learning process and the existing e-learning systems. It could teach the students more than just certain contents: it could teach them how to learn, by selecting and sharing the adequate information in each moment.

In this point, we will remark some pedagogical characteristics of e-learning computer models which are known to improve the learning process. For each of these characteristics we propose a feature that can be implemented by using New Technologies.

Pedagogical Characteristic 1

Dealing with *information of different sources* will allow the students seeing different points of view of the same realities, making easy its understanding and its conservation in mind.

New Technologies Feature 1

In the Institutional Memory of a *Knowledge Management System*, we will find all the information concerning every thematic unit, different levels and its associated tasks. The fact of being able to solve different tasks and having the access into information of different sources allows the learner to acquire the information by different means, so that his knowledge will be more complete and everlasting.

Pedagogical Characteristic 2

An e-learning model should provide an *individual attention*, taking into account the student's preferences about learning strategies, different kind of materials, their previous knowledge, etc.

New Technologies Feature 2

In order to get it, some *intelligent agents* can take charge of selecting and showing, in any case, the suitable information (from the information repository) according to the preferences and level of each student. These agents can perform different tasks, and divide them between the users' computers and the server where the Institutional Memory is stored.

Pedagogical Characteristic 3

An e-learning model must *facilitate the students all the available information*, in different formats and coming from different sources, for the students to learn how to choose the most relevant elements for their learning.

New Technologies Feature 3

To reach this, it can be used a *global ontology*, establishing a classification in levels and the relationships between the available information, managed trough the Knowledge Management System.

Pedagogical Characteristic 4

Computer e-learning models ought to *propose this apprenticeship by means of works and problems to solve*, so that the students knowledge grows as they go on with the resolution of their works.

New Technologies Feature 4

It is necessary to establish much different kind of works, at different levels, for each unit the student must prepare. The tasks and the available information as well, will be founded in the Institutional Memory organized under *ontology*, making easy the access to the relevant information at any moment. Carrying out the tasks, the learner will build his own knowledge (Nonaka, 1995).

Pedagogical Characteristic 5

An e-learning model should join *strategies to get and raise the student's motivation* and encourage its inquisitiveness, relating the available information with its interest, proposing the possibility of explore deeper the same or related subjects and using the computer games strategies that give rise to investigate.

New Technologies Feature 5

When the contents of the course are part of an Institutional Memory, the existence of a global ontology can facilitate the display of the elements remarking the connections between them. Besides, as alleged previously, the use of intelligent agents allows us to show these connections according to the individual preferences. The *strategies utilized in computer games, including the apprenticeship trough the action* will help to attract and maintain the student's interest.

FUTURE WORKS

Some prototypes for the aspects mentioned in the previous point have been developed in our research laboratory for testing the proposed features (Pedreira, 2004a, 2004b, 2005a, 2005b). Each of them has reached quite good results. These approximations show that the use of New Technologies on education allows the students to extend or improve their problem-solving methods and their abilities to transfer knowledge (Friss de Kereki, 2004). After these first approaches, we are working on the joint of the prototypes and their enlargement with some characteristics that have not still been tested.

CONCLUSION

In this article we suggest several features that e-learning systems should have in order to improve online learning, which can be achieved by using New Technologies. In short, we propose a computer model based on a Knowledge Management System which, by using a global ontology, maintains the highest quantity of relationships between the available information and its classification at different levels. By means of this support of knowledge, apprenticeship can be established by means of task proposal, based on computer game strategies. Using the philosophy of intelligent agents, these systems can interact with the students showing them the information according to their preferences, in order to motivate them and to stimulate their capacity of raising questions.

REFERENCES

Friss de Kereki, I. (2004). Modelo para la creación de entornos de aprendizaje basados en técnicas de Gestión de Conocimiento. Doctoral Thesis. Politechnical University of Madrid.

Martínez, J. (2002). Contenidos en e-learning: el rey sin corona. Retrieved february 13, 2003, from http://www.gestiondelconocimiento.com.

Nonaka, I.& Takeuchi, H. (1995). The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation. New York: Oxford University Press.

Pedreira, N., Rabuñal, J. (2005a) La Gestión del Conocimiento como Instrumento de e-Learning. E-World, 129-141. Fundación Alfredo Brañas. Santiago de Compostela.

Pedreira, N., Dorado, J., Rabuñal, J., Pazos, A. (2005b) Knowledge Management as the Future of E-learning. Encyclopedia of Distance Learning. Vol 1 - Distance Learning Technologies and Applications, 1189-1194. Idea Group Reference.

Pedreira, N., Dorado, J., Rabuñal, J., Pazos, A., Silva, A. (2004a). A Model of Virtual 'Learning to Learn'. Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04), 838-839. IEEE Computer Society Washington, DC, USA

Pedreira, N., Dorado, J., Rabuñal, J., Pazos, A., Silva, A. (2004b). Knowledge Management and Interactive Learning. Lecture Notes in Computer Science. Vol 3257, 481-482. Springer-Verlag Heidelberg

Piaget, J.(1999). De la pedagogía. Buenos Aires: Paidós.

Wiig, K. M. (1995). Knowledge Management Methods. Arlington: Schema Press, Ltd.

WorldWideLearn (2007). Benefits of E-Learning. World Wide Learn, "The World's Premier Online Directory of Education". Retrieved march 11, 2007, from http://www.worldwidelearn.com.

KEY TERMS

Best Practice: A management idea which asserts that there is a technique, method, process, activity, incentive or reward that is more effective at delivering a particular outcome than any other technique, method or process.

Computer Game: A video game played on a personal computer, rather than on a video game console or arcade machine.

Computer Model: A computer program that attempts to simulate an abstract model of a particular system.

E-Learning: Learning that is accomplished over the Internet, a computer network, via CD-ROM, interactive TV, or satellite broadcast

Intelligent Agent: A real time software system that interacts with its environment to perform non-repetitive computer-related tasks.

Knowledge Management: The collection, organization, analysis, and sharing of information held by workers and groups within an organization.

Learn to Learn: In this context, learn to manage (select, extract, classify) the great amount of information existing in actual society, in order to identify real and significant knowledge.

New Technologies: In this context, Computer, Information and Communication Technologies.

Virtual: Not physical.

Emerging Applications in Immersive Technologies

Darryl N. Davis
University of Hull, UK

Paul M. Chapman
University of Hull, UK

INTRODUCTION

The world of Virtual Environments and Immersive Technologies (Sutherland, 1965) (Kalawsky, 1993) are evolving quite rapidly. As the range and complexity of applications increases, so does the requirement for intelligent interaction. The now relatively simple environments of the OZ project (Bates, Loyall & Reilly, 1992) have been superseded by Virtual Theatres (Doyle & Hayes-Roth, 1997) (Giannachi, 2004), Tactical Combat Air (Jones, Tambe, Laird & Rosenbloom, 1993) training prototypes and Air Flight Control Simulators (Wangermann & Stengel, 1998).

This article presents a brief summary of present and future technologies and emerging applications that require the use of AI expertise in the area of immersive technologies and virtual environments. The applications are placed within a context of prior research projects.

BACKGROUND

Visualisation is defined as the use of computer-based, interactive visual representations of data to amplify cognition. The much cited process driven visualisation pipeline proposed by Upson et al (1989) is shown in Figure 1. Upson and his colleagues define three processes consisting of filtering, mapping and rendering the data. The image presented allows the user to draw some inference and gain insight into the data.

The *Filter* process is when data of interest are derived from the raw input data; for example, an interpolation of scattered data onto a regular grid. This data is then *Mapped* into geometric primitives that can be then be *Rendered* and displayed as an image to the user. The user may then gain an improved understanding and greater insight into the original raw data. The type of data and application area heavily influence the nature of the mapping process. That is, choosing the actual visualisation technique that we are going to use. For example, if the data consisted of 1D scalar data, then a simple line graph can be used to represent the data. If the filtered data consists of 3D scalar data, then some form of 3D isosurfaces or direct volume rendering technique would be more appropriate. Through the various specifications and conceptualisations of the filter-map pipeline above, we would propose an ontology that describes the relationships between data type and mapping processes that facilitates the automatic selection of visualisation techniques based on the raw data type. As the applications become more sophisticated the visualisation process can make use of the data ontology to drive AI controlled characters and agents appropriate for the application and data.

A starting place for this can be seen in the area of believable agents (Bates, Loyall & Reilly, 1992) where the research ranges from animation issues to models of emotion and cognition, annotated environments. Innovative learning environments and Animated Pedagogical Agents (Johnson, Rickel & Lester, 2000) provide further areas for development, as do industrial

Figure 1. Upson et al's visualisation pipeline
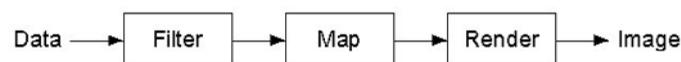
applications, for example Computer Numerical Control (CNC) milling operations virtual training prototype system (Lina, Yeb, Duffy & Suc, 2002). Teaching environments using multiple interface devices in virtual reality include Steve (Soar Training Expert for Virtual Environments) that supports the learning process (Rickel & Johnson, 1999) with collaborators including Lockheed Martin AI Center. SOAR (Laird, Hucka & Huffman, 1991) has also been used for training simulation in air combat (TacAir-Soar) (Jones, Tambe, Laird & Rosenbloom, 1993). This autonomous system for modelling the tactical air domain brings together areas of AI research covering cognitive architectures Human Behavior Representation (HBR) / Computer Generated Forces (CGF). SOF-Soar: (Special Operations Forces Modeling) (Tambe, Johnson, Jones, Koss, Laird, Rosenbloom and Schwamb, 1995) uses the same underlying framework and methods for behavior generation based on the Soar model of human cognition, each entity is capable of autonomous, goal-directed decision-making, planning, and reactive, real-time behavior. As the world of digital media expands emerging applications will draw on the worlds of Virtual Theatres (Giannachi, 2004), interactive storyline games and others forms of entertainment (Wardrip-Fruin & Harrigan, 2004) to enhance the visualisation experience, especially where virtual worlds involving human artifacts and past and current civilisations are involved.

SWARM INTELLIGENCE FOR VISUALISATION

Understanding the behaviour of biological agents in their natural environment is of great importance to ethologists and biologists. Where these creatures move in large numbers is a challenge for orthodox visualisation.

Working with marine biologists, a 3D model of large numbers of swarming krill (Figure 2) has been created. The model augments the classic swarming functions of separation, alignment and cohesion outlined by Reynolds (1987). The generated 3D model allows cameras to be placed on individual krill in order to generate an in-swarm perspective. New research on Antarctic krill (Tarling & Johnson, 2006) reveals that they absorb and transfer more carbon from the Earth's surface than was previously understood. Scientists from the British Antarctic Survey (BAS) and Scarborough Centre of Coastal Studies at the University of Hull discovered that rather than doing so once per 24 hours, Antarctic krill 'parachute' from the ocean surface to deeper layers several times during the night. In the process they inject more carbon into the deep sea when they excrete their waste than had previously been understood. Our objective has been to provide marine biologists with a visualisation and statistical tool that permits them to change a number of parameters within the krill marine environment and examine the effects of those changes over time. The software can also be used as a teaching tool for the classroom at varying academic levels.

Figure 2. 3D krill and sample 3D swarm

The marine biologist may modify parameters relating to an individual krill's field of view, foraging speed, collision avoidance, exhaustion speed, desire for food etc. The researcher may also modify more global parameters relating to sea currents, temperature and quantity and density of algae (food) etc.

Upon execution, the biologist may interact with the model by changing the 3D viewpoint and modifying the time-step which controls the run speed of the model. Krill states are represented using different colours. For example, a red krill represents a starved unhealthy krill, green means active and healthy, blue represents a digestion period where krill activity is at a minimal. Recent advances in processor and graphics technology means that these sorts of simulations are possible using high specification desktop computers. Marine biologists have been very interested in seeing how making minor changes to certain variables, such as field of view for krill, can have major consequences to the flocking behaviour over time of the entire swarm.

PARAGLIDING SIMULATOR

The Department of Computer Science at the University of Hull have recently developed the world's first ever paragliding simulator (SimVis, 2007). The system provides a paragliding pilot with a virtual reality immersive flying experience (Figure 3). As far as the user is concerned, they are flying a real paraglider. They sit in a real harness and all physical user inputs are the same as real life. Visuals are controlled via a computer and displayed to the user via a head-tracked helmet mounted display. The simulator accurately models winds (including thermals and updrafts), photorealistic terrain and other computer controlled AI pilots.

Figure 4 shows a typical view from the paragliding simulator. To the right of the image, the user can see four paragliding pilots circling a thermal. It is in the user's interests to fly to this region to share in the uplift, gaining altitude and therefore flight time.

This prototype is being developed along the lines of SOF-Soar: (Special Operations Forces Modeling) (Tambe, Johnson, Jones, Koss, Laird, Rosenbloom and Schwamb, 1995). It requires an expert tutoring system encompassing the knowledge of expert pilots. Like some virtual learning environments it needs to build a trainee profile from a default and adapt to the needs

Figure 3. VR paragliding simulator

of the novice flyer. For example the system can create an AI Pilot flying directly at the user forcing the user to practise collision avoidance rules. If pilots collide in the simulator, both pilots will become wrapped in each others canopies and they will plummet down to the earth and die. As virtual fly-time accumulates the system needs to adapt to the changing profile of the user. An expert system could be used to force the user to make certain flight manoeuvres that test the user's knowledge of CAA air laws. For example, the AI system may decide that the user is an advanced pilot due to their excellent use of thermal updrafts etc. The system therefore works out how to put our pilot into a compromising situation that would test their skills and ability such as plotting a collision course with our pilot when they are flying alongside a cliff edge. If our pilot is a novice, then the system would present simpler challenges such as basic collision avoidance. This is an advanced knowledge engineering project that blends traditional AI areas such as knowledge engineering with more nouvelle fields such as agent based reactive and cognitive architectures and state of the art visualisation and immersive technologies.

Figure 4. Real-time pilot's view from the paragliding simulator including other AI controlled pilots

CONCLUSION

This article suggests that cognitive science and artificial intelligence have a major role to play in emerging interface devices that require believable agents. As virtual environments and immersive technologies become ever more sophisticated, the capabilities of the interacting components will need to become smarter, making use of artificial-life, artificial intelligence and cognitive science. This will include the simulation of human behaviour in interactive worlds whether in the modelling of the way data and information is manipulated or in the use of hardware devices (such as the paraglider) within a virtual environment.

REFERENCES

Bates, J., Loyall, A.B. & Reilly, W.S. (1992). An Architecture for Action, Emotion, and Social Behavior. Tech. Report CMU-CS-92-142, School of Computer Science, Carnegie Mellon University, Pittsburgh.

Doyle, P. & Hayes-Roth, B. (1997). Guided exploration of virtual worlds. In Network and Netplay: Virtual Groups on the Internet. F. Sudweeks, Ed. MIT Press: Cambridge, MA.

Giannachi, G. (2004). Virtual Theatres: An Introduction. London: Routledge Press.

Hayes-Roth, B., Brownston, L. & van Gent, R. (1995). Multiagent collaboration in directed improvisation. Proc. 1st Int. Conf. on Multi-Agent Systems, San Francisco; Reprinted in Readings in Agents, M. Huhns and M. Singh, Eds. Morgan-Kaufmann: San Francisco.

Johnson, W.L, Rickel, J.W. & Lester, J.C. (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. International Journal of Artificial Intelligence in Education 11, 47-78.

Jones, R. Tambe, M., Laird, J. & Rosenbloom, P. (1993). Intelligent automated agents for flight training simulators. Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation. University of Central Florida. IST-TR-93-07.

Kalawsky, R. S. (1993). The Science of Virtual Reality and Virtual Environments: A Technical, Scientific and Engineering Reference on Virtual Environments. Addison-Wesley, Wokingham, England ; Reading, Mass.

Laird, J., Hucka, M. & Huffman, S. (1991). An analysis of Soar as an integrated architecture. SIGART Bulletin, 2, 85-90.

Lina, F., Yeb, L., Duffy, V.G. and Suc, C.-J. (2002). Developing virtual environments for industrial training. Information Sciences, 140(1-2), 153-170.

Reynolds, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioral Model. Computer Graphics, 21(4), 25-34.

Rickel, J. & Johnson, W.L. (1999). Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control. Applied Artificial Intelligence, 13, 343-382.

SimVis (2007) Virtual Paragliding Project, SimVis Research Group. Web: www.dcs.hull.ac.uk/simvis/projects/paraglider

Sutherland, I. E. (1965). The Ultimate Display. Proceedings of IFIP 65, (2), 506-508

Tambe, M., Johnson, W. L., Jones, R. M., Koss, F., Laird, J. E., Rosenbloom, P. S. & Schwamb, K. (1995). Intelligent Agents for Interactive Simulation Environments. AI Magazine, 16(1).

Tarling, G. & Johnson, M. (1996). Satiation gives krill that sinking feeling. Current Biology, 16(3), 83-84.

Upson, C., Jr, T. F., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R. & van Dam, A. (1989). The Application Visualisation System: A Computational Environment for Scientific Visualisation. IEEE Computer Graphics and Applications 9(4), 30-42.

Wangermann, J.P. & R. F. Stengel, R.F. (1998). Principled Negotiation Between Intelligent Agents: A Model for Air Traffic Management. Artificial Intelligence in Engineering, 12(3), 177-187.

Wardrip-Fruin, N. & Harrigan, P. (Eds.) (2004). First Person: New Media as Story, Performance, and Game. MIT Press.

KEY TERMS

Air Flight Control: A service provided by ground-based controllers who direct aircraft on the ground and in the air. A controller's primary task is to separate certain aircraft — to prevent them from coming too close to each other by use of lateral, vertical and longitudinal separation. Secondary tasks include ensuring orderly and expeditious flow of traffic and providing information to pilots, such as weather, navigation information and NOTAMs (Notices to Airmen)

Flocking: A computer model of coordinated animal motion such as bird flocks and fish schools. Typically based on three dimensional computational geometry of the sort normally used in computer animation or computer aided design.

Knowledge Engineering: Knowledge engineering is a field within artificial intelligence that develops knowledge-based systems. Such systems are computer programs that contain large amounts of knowledge, rules and reasoning mechanisms to provide solutions to real-world problems. A major form of knowledge-based system is an expert system, one designed to emulate the reasoning processes of an expert practitioner (i.e. one having performed in a professional role for very many years).

Virtual and Immersive Environments: Virtual environments coupled with immersive technologies provide the sensory experience of being in a computer generated, simulated space. They have potential uses in applications ranging from education and training to design and prototyping.

Virtual Theatres: The concept of "Virtual Theatre" is vague and there seems to be no commonly accepted definition of the term. It can be defined as a virtual world inhabited by autonomous agents that are acting and interacting in an independent way. These agents may follow a predetermined manuscript, or act completely on their own initiative.

Emulating Subjective Criteria in Corpus Validation

Ignasi Iriondo
Universitat Ramon Llull, Spain

Santiago Planet
Universitat Ramon Llull, Spain

Francesc Alías
Universitat Ramon Llull, Spain

Joan-Claudi Socoró
Universitat Ramon Llull, Spain

Elisa Martínez
Universitat Ramon Llull, Spain

INTRODUCTION

The use of speech in human-machine interaction is increasing as the computer interfaces are becoming more complex but also more useable. These interfaces make use of the information obtained from the user through the analysis of different modalities and show a specific answer by means of different media. The origin of the multimodal systems can be found in its precursor, the "Put-That-There" system (Bolt, 1980), an application operated by speech and gesture recognition.

The use of speech as one of these modalities to get orders from users and to provide some oral information makes the human-machine communication more natural. There is a growing number of applications that use speech-to-text conversion and animated characters with speech synthesis.

One way to improve the naturalness of these interfaces is the incorporation of the recognition of user's emotional states (Campbell, 2000). This point generally requires the creation of speech databases showing authentic emotional content allowing robust analysis. Cowie, Douglas-Cowie & Cox (2005) present some databases showing an increase in multimodal databases, and Ververidis & Kotropoulos (2006) describe 64 databases and their application. When creating this kind of databases the main arising problem is the naturalness of the locutions, which directly depends on the method used in the recordings, assuming that they must be controlled without interfering the authenticity

of the locutions. Campbell (2000) and Schröder (2004) propose four different sources for obtaining emotional speech, ordered from less control but more authenticity to more control but less authenticity: i) natural occurrences, ii) provocation of authentic emotions in laboratory conditions, iii) stimulated emotions by means of prepared texts, and iv) acted speech reading the same texts with different emotional states, usually performed by actors.

On the one hand, corpora designed to synthesize emotional speech are based on studies centred on the listener, following the distinction made by Schröder (2004), because they model the speech parameters in order to transmit a specific emotion. On the other hand, emotion recognition implies studies centred on the speaker, because they are related to the speaker emotional state and the parameters of the speech. The validation of a corpus used for synthesis involves both kinds of studies: the former since it will be used for synthesis and the latter since recognition is needed to evaluate its content. The best validation system is the selection of the valid utterances[1] of the corpus by human listeners. However, the big size of a corpus makes this process unaffordable.

BACKGROUND

Emotion recognition has been an interesting research field in human-machine interaction for long, as can be

observed in Cowie et al. (2001). Some studies have been carried out to observe the influence of emotion in speech signals like the work presented by Rodríguez et al. (1999), but more recently, due the increasing power of modern computers that allows the analysis of huge amount of data in relatively small time lapses, machine learning techniques have been used to recognise emotions automatically by using labelled expressive speech corpora. Most of these studies have been centred on few algorithms and little sets of parameters.

However, recent works have performed more exhaustive experiments testing different machine learning techniques and datasets, as the described by Oudeyer (2003). All this kind of studies had the goal of achieving the best possible recognition rate obtaining, in many cases, better results than those obtained in subjective tests ((Oudeyer, 2003), (Planet, Morán & Formiga, 2006), (Iriondo, Planet, Socoró & Alías, 2007)). Nevertheless, many differences can be found when analyzing the results obtained from objective and subjective classifications and, to our knowledge, there are not studies with the goal of emulating these subjective criteria before those carried out by Iriondo, Planet, Alías, Socoró & Martínez (2007).

VALIDATION OF AN EXPRESSIVE SPEECH CORPUS BY MAPPING SUBJECTIVE CRITERIA

The creation of a speech corpus with authentic emotional content is one of the most important challenges in the study of expressive speech. Once the corpus is recorded, a validation process is required to prune those utterances that show distinct emotion to their label. This article is based on the work exposed by Iriondo, Planet, Alías, Socoró & Martínez (2007) and presents the production of an expressive speech corpus in Spanish with the goal of being used in a synthesis system, validating it by pruning automatically "bad" utterances emulating the criteria of human listeners.

The Production of the Corpus

The recording of the corpus has been carried out by a female professional speaker. There is a high consensus in the scientific community for obtaining emotional

speech by means of this strategy for synthesis purposes (Cowie et al., 2005), although other authors argue in favor of constructing enormous corpora gathered from recordings of the daily life (Campbell, 2005). For the design of texts semantically related to different expressive styles, we have made use of an existing textual database of advertisements extracted from newspapers and magazines. Based on a study of the voice in the audio-visual publicity (Montoya, 1998), five categories of the textual corpus have been chosen and the most suitable emotion/style has been assigned to them: New technologies (neutral-mature), education (joy-elation), cosmetic (style sensual-sweet), automobiles (aggressive-hard) and trips (sad-melancholic). The recorded database has 4638 sentences and it is 5 hours 12 minutes long.

From these categories, a set of sentences has been chosen by means of a greedy algorithm (François & Boëffard, 2002) that has allowed us to select phonetically balanced sentences. In addition to looking for a phonetic balance, phrases that contain foreign words and abbreviations have been discarded because they difficult the automatic process of phonetic transcription and labeling.

The corpus has been segmented in phrases and then in phonemes by means of a semiautomatic process based on a forced alignment with Hidden Markov Models.

Acoustic Analysis

Cowie et al. (2001) show how prosodic features of speech (fundamental frequency (F0), energy, duration of phones, and frequency of pauses) are related to vocal expression of emotion. The analysis of F0 performed in this work is based on the result of the pitch marks algorithm described by Alías, Monzo & Socoró (2006). This system can assign marks over the whole signal, interpolating values from the neighbour phonemes in unvoiced segments and silences. Energy is measured with 20 ms rectangular windows and 50% of overlap, computing the mean energy in decibels (dB) every 10 ms. Also, rhythm parameters have been incorporated using the z-score as a means to analyze the temporal structure of speech (Schweitzer & Möbius, 2003). Moreover, for each utterance two parameters relating the number of pauses per time unit and the percentage of silence respect to the total time are considered.

Subjective Test

A subjective test allows validating the expressiveness of a corpus of acted speech from a user's point of view. Nevertheless, an extensive evaluation of the complete corpus would be very tedious due the big amount of elements in it. For this reason, only a 10 percent of the corpus utterances have been selected for this test. A forced answer test has been designed using the TRUE platform (Planet, Iriondo, Martínez, & Montero, 2008) with the question: *What emotional state do you recognize from the voice of the speaker in this phrase?* The possible answers are the 5 emotional styles of the corpus and one more option *Don't know / Another* (Dk/A) in order to minimize biasing the results due to confusing cases. The addition of this option has the risk of allowing some users to use excessively this answer to accelerate the end of the test (Navas, Hernáez & Luengo, 2006). However, this effect has not been observed in this experiment. The evaluators were 30 volunteers with a quite heterogeneous profile.

The achieved average classification accuracy in the subjective test is 87%. The test also reveals that sad style (SAD) is the best rated (98.5% in average). The second and third best rated styles are sensual (SEN) (87.2%) and neutral (NEU) (86.1%), followed by happy (HAP) (81.9%) and aggressive (AGR) (81.6%). Aggressive and happy styles are often confused between them. Moreover, sensual is slightly misclassified as sad or neutral. The Dk/A option is hardly used, although it is more present in neutral and sensual than in the rest of styles.

To decide if utterances are not optimally performed by the speaker, two simple rules have been created from the subjective test results by empirically adjusting two thresholds. These rules remove utterances whose identification percentage is lower than 50% or with a Dk/A percentage larger than 12%. There are 33 out of the 480 utterances of the subjective test that satisfy at least one rule.

Statistical Analysis, Datasets and Supervised Classification

Iriondo, Planet, Socoró & Alías (2007) present an experiment of emotion recognition covering different datasets and algorithms. The experiment is done with the same corpus that is being considered in this article. Each utterance is defined by 464 attributes representing the speech signal characteristics but this first dataset is divided into different subsets to reduce its dimensionality. The experiments show almost the same results in the full dataset and in a dataset reduced to 68 parameters, so the reduced dataset is being used in this work. In this dataset, the prosody of an utterance is represented by the vectors of logarithmic F0, energy in dB and normalized durations. For each sequence, the first derivative is also calculated. Some statistics are obtained from these sequences: mean, variance, maximum, minimum, range, skewness, kurtosis, quartiles, and interquartilic range. Thus, 68 parameters by utterance are computed, considering both parameters related to the pausing previously described.

In the referenced work, twelve machine learning algorithms are tested considering different datasets. All the experiments are carried out using Weka software (Witten & Frank, 2005) by means of ten-fold cross-validation. Very high recognition rates are achieved as in other previously referenced works: SMO (SVM of Weka) obtained the best results (~97%) followed by Naïve-Bayes (NB) with 94.6% and J48 (Weka Decision Tree based on C4.5) with 93.5%, considering the average of all the results. The conclusion is that, in general, the styles of the developed speech corpus can be clearly differentiated. Moreover, the results of the subjective test showed a good authenticity of the expressive speech content. However, going one step further by developing a method to validate each utterance of the corpus following subjective criteria and not only the automatic classification from the space of attributes is considered necessary.

Subjective Based Attribute Selection

The proposed approach to find the optimum classifier schema able to follow the subjective criteria consists on the development of an attribute selection method guided by the results obtained in the subjective test. Once the best schema is found it will be applied to the whole corpus in order to generate a list of candidate utterances to be pruned. Two methods for attribute selection have been developed in order to determine the subset of attributes that allows a better mapping of the subjective test results. As previously mentioned,

the original set has 68 attributes per utterance, so an exhaustive exploration of subsets is not viable and greedy search procedures will be used. On the one hand, a Forward Selection (FW) process is chosen, which starts without any attribute and adds one at a time and, on the other hand, the Backward Elimination (BW) technique is considered, which starts from the full set and deletes one attribute at a time. At each step, the classifier is tested with the 480 utterances that are considered in the subjective test, being previously trained with the 4158 remaining utterances. The wrong classified cases take part on the evaluation process of the involved subset of attributes. The novelty of this process is to use a subjective-based measure to evaluate the expected performance of the subset at each iteration. The used measure is the F1-score computed from the precision and the recall of the wrong classified utterances compared with the 33 utterances rejected by the subjective test.

Results

The process consists of six iterations: one per algorithm (SMO, NB and J48) and attribute selection technique (FW and BW). SMO algorithm obtains practically the same F1-score (~ 0.50) in both FW and BW attribute selection techniques. The results for both NB are also similar between them (F1 ≈ 0.43). The main difference is for J48 that obtains better F1 with FW (0.45) than with BW (0.37) process. Moreover, SMO-FW seems to be the most stable configuration as it achieves almost the same result (F1 = 0.49) with a broad range of attribute subsets (the results are very similar with a range of attributes between 18 and 35). Results show that J48-FW has the best recall (18/33), which implies the highest number of coincidences; however the precision measure is quite low (18/51), indicating an excessive number of general misclassifications.

FUTURE TRENDS

Future work will consist on applying this automatic process over the full corpus. For instance, a ten-fold cross validation would be a good technique to cover the whole corpus. The misclassified utterances would be candidate to be pruned. A first approach would consist on running different classifiers and selecting

the final candidates by a stacking technique (Witten & Frank, 2005). Also, we will evaluate the suitability of the proposed method by performing acoustic modeling of the resulting emotional speech after pruning with respect to the results obtained with the whole corpus. A lower error on the prosodic estimation could confirm the hypothesis that it is advisable to eliminate the bad utterances of the corpus previously.

CONCLUSION

This article exposes the need of an automatic validation of an expressive speech corpus due to the impossibility of carrying out a subjective test in a large corpus. Also, an approach to achieve this goal has been presented, performing a subjective test with 30 subjects and 10 percent of the utterances, approximately. The result of this test has shown that some utterances are perceived with a dissimilar or poor expressive content with respect to their labeling. The proposed automatic classification tries to learn from the result of the subjective test in order to generalize the solution to the rest of the corpus, by means of a suitable attribute selection carried out by two different strategies (Forward Selection and Backward Elimination) using the F1-measure computed taking into account the misclassifications resulting from the subjective test as a reference.

REFERENCES

Alías, F., Monzo, C. & Socoró, J. C. (2006). A pitch marks filtering algorithm based on restricted dynamic programming. *InterSpeech2006 - International Conference on Spoken Language Processing*, pp. 1698-1701. Pittsburgh.

Bolt, R. A. (1980). "Put-that-there": Voice and gesture at the graphics interface. *Proceedings of the 7th annual conference on Computer graphics and interactive techniques* pp. 262-270. Seattle, Washington: ACM Press.

Campbell, N. (2000). Databases of emotional speech. *Proceedings of the ISCA Workshop*, pp. 34-38.

Campbell, N. (2005). Developments in corpus-based speech synthesis: approaching natural conversational speech. *IEICE - Trans. Inf. Syst. , E88-D* (3), 376-383.

Cowie, R., Douglas-Cowie, E. & Cox, C. (2005). Beyond emotion archetypes: databases for emotion modelling using neural networks. *Neural Networks , 18*, 371-388.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., et al. (2001). Emotion Recognition in Human-Computer Interaction. *IEEE Signal Processing Magazine , 18* (1), 32-80.

François, H. & Boëffard, O. (2002). The greedy algorithm and its application to the construction af a continuous speech database. *Proceedings of LREC*, *5*, pp. 1420-1426.

Iriondo, I., Planet, S., Alías, F., Socoró, J.-C. & Martínez, E. (2007). Validation of an expressive speech corpus by mapping automatic classification to subjective evaluation. *Lecture Notes on Computer Science. Computational and Ambient Intelligence, 9th International Work-Conference on Artificial Neural Networks. 4507.* San Sebastián: Springer-Verlag.

Iriondo, I., Planet, S., Socoró, J.-C. & Alías, F. (2007). Objective and subjective evaluation of an expressive speech corpus. *ISCA Tutorial and Research Workshop on NOn LInear Speech Processing.* Paris.

Montoya, N. (1998). El papel de la voz en la publicidad audiovisual dirigida a los niños. *Zer. Revista de estudios de comunicación ,* 161-177.

Navas, E., Hernáez, I. & Luengo, I. (2006). An objective and subjective study of the role of semantics and prosodic features in building corpora for emotional TTS. *IEEE Transactions on Audio, Speech and Language Processing* (14).

Oudeyer, P.-Y. (2003). The production and recognition of emotions in speech: features and algorithms. Special issue on Affective Computing. *International Journal of Human Computer Interaction , 59* (1-2), 157-183.

Planet, S., Iriondo, I., Martínez, E. & Montero, J. A. (2008). TRUE: an online testing platform for multimedia evaluation. In *Proceedings of the Second International Workshop on EMOTION: Corpora for Research and Affect at the 6th Conference on Language Resources & Evaluation* (LREC 2008). Marrakech, Morocco.

Planet, S., Morán, J. A. & Formiga, L. (2006). Reconocimiento de emociones basado en el análisis de la señal de voz parametrizada. *Sistemas e Tecnologias*

de Informação no Espaço Ibérico, *II*, pp. 837-854. Esposende, Portugal.

Rodríguez, A., Lázaro, P., Montoya, N., Blanco, J. M., Bernadas, D., Oliver, J. M., et al. (1999). Modelización acústica de la expresión emocional en el español. *Procesamiento del Lenguaje Natural* (25), 152-159.

Schröder, M. (2004). *Speech and emotion research: An overview of research frameworks and a dimensional approach to emotional speech synthesis.* PhD thesis, Saarland University.

Schweitzer, A. & Möbius, B. (2003). On the structure of internal prosodic models. *Proceedings of the 15th ICPhS*, pp. 1301-1304. Barcelona.

Ververidis, D. & Kotropoulos, C. (2006). Emotional speech recognition: Resources, features and methods. *Speech Communication , 48* (9), 1162-1181.

Witten, I. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

KEY TERMS

Backward Elimination Strategy: Greedy attribute selection method that evaluates the effect of removing one attribute from a dataset. The attribute that improves the performance of the dataset when it is deleted is chosen to be removed for the next iteration. Process begins with the full set of attributes and stops when no attribute removing improves performance.

Decision Trees: Classifier consisting on an arboreal structure. A test sample is classified by evaluating it in each node, starting at the top one and choosing a specific branch depending on this evaluation. The classification of the sample is the class assigned in the bottom node.

F1-Measure: The F1-measure is an approach of combining the precision and recall measures of a classifier by means of an evenly harmonic mean of both them. Its expression is F1-measure = (2×precision×recall) / (precision+recall).

Forward Selection Strategy: Greedy attribute selection method that evaluates the effect of adding one attribute to a dataset. The attribute that improves

the performance of the dataset is chosen to be added to the dataset for the next iteration. Process begins with no attributes and stops when adding new attributes provides no performance improvement.

Greedy Algorithm: Algorithm -usually applied to optimization problems- based on the idea of finding a global solution for a problem (despite of not being the optimal one) by choosing locally optimal solutions in different iterations.

Naïve-Bayes: Probabilistic classifier based on Baye's rule that assumes that all the pairs parameter-value that define a case are independent.

Precision: Measure that indicates the percentage of correctly classified cases of one class with regard to the number of cases that are classified (correctly or not) as members of that class. This measure says if the classifier is assuming as members of one specific class cases from other different classes.

Recall: Measure that indicates the percentage of correctly classified cases of one class with regard to the total number of cases that actually belong to this class. This measure says if the classifier is ignoring cases that should be classified as members of one specific class when doing a classification.

SVM: Acronym of Support Vector Machines. SVM are models able to distinguish members of classes whose limits are not lineal. This is possible by a non-linear transformation of input data mapping it into a higher-dimensionality space where data can be easily divided by a maximum margin hyperplane.

ENDNOTE

[1]   Considering as valid utterances those with the adequate expressiveness.

Energy Minimizing Active Models in Artificial Vision

Gloria Bueno García
University of Castilla – La Mancha, Spain

Antonio Martínez
University of Castilla – La Mancha, Spain

Roberto González
University of Castilla – La Mancha, Spain

Manuel Torres
University of Castilla – La Mancha, Spain

INTRODUCTION

Deformable models are well known examples of artificially intelligent system (AIS). They have played an important role in the challenging problem of extracting useful information about regions and **areas of interest** (ROIs) imaged through different modalities. The challenge is also in extracting boundary elements belonging to the same ROI and integrate them into a coherent and consistent model of the structure. Traditional low-level image processing techniques that consider only local information can make incorrect assumptions during this integration process and generate unfeasible object boundaries. To solve this problem, deformable models were introduced (Ivins, 1994), (McInerney, 1996), (Wang, 2000). These AI models are currently important tools in many scientific disciplines and engineering applications (Duncan, 2000).

Deformable models offer a powerful approach to accommodate the significant variability of structures within a ROI over time and across different individuals. Therefore, they are able to segment, match and track images of structures by exploiting (bottom-up) constraints derived from the image data together with (top-down) *a priori* knowledge about the location, size, and shape of these structures.

The mathematical foundations of deformable models represent the confluence of geometry, physics and approximation theory. Geometry serves to represent object shape, physics imposes constraints on how the shape may vary over space and time, and optimal approximation theory provides the formal mechanisms for fitting the models to data. The physical interpretation views deformable models as elastic bodies which respond to applied force and constraints.

BACKGROUND

The deformable model that has attracted the most attention to date is the **active contour model** (ACM), well-known as **snakes**, presented by Kass *et al*. (Kass, 1987), (Cootes & Taylor, 1992). The mathematical basis present in snake models is similar to all deformable models, which are based on **energy minimizing** techniques.

Recently, there has been an increasing interest in level set or geodesic segmentation methods, introduced in (Osher & Sethian, 1988), (Malladi, 1995) and (Caselles, 1997). Level set approach involves solving the ACM minimization problem by the computation of minimal distances curve. This method allows topological changes within the ROIs and extension to 3D. Therefore, for some applications it is an improvement on classical ACM.

Other approaches to deformable model are those based on dynamic models or physically based techniques, for example superquadrics (Terzopoulos, 1991) and the finite element model (FEM) (Pentland, 1991). The FEM accurately describes changes in position, orientation and shape. The FEM can be used to solve fitting, interpolation or correspondence problems. In the FEM, interpolation functions are developed that allow continuous material properties, such as mass

and stiffness, to be integrated across the ROIs. This last property makes them different from the previous models and therefore more suitable for some artificial vision applications.

The next sections contain a brief introduction to the mathematical foundations of deformable models.

ENERGY MINIMIZING DEFORMABLE MODELS

Geometrically, an active contour model is a parametric contour embedded in the image plane $(x, y) \in \mathbb{R}^2$. The dynamic contour is represented as a time-varying curve, $v(s,t) = (x(s,t), y(s,t))$, where x and y are the coordinate functions and $s \in [0, 1]$ is the parametric domain. The curve evolves until the ROI, subject to constraints from a given image $I(x, y)$, reaches an equilibrium. Thus, initially a curve is set around the ROI that, *via* minimization of an energy functional, moves normal to itself and stops at the boundary of the ROI. The energy functional is defined as:

$$E_{snake}(s,t) = \int_0^1 \left[E_{internal}(v(s,t)) + E_{ext\_potential}(v(s,t)) \right] ds$$

(1)

The first term, $E_{internal}$, represents the internal energy of the spline curve due to mechanical properties of the contour, stretching and bending. It is a sum of two components, the elasticity and rigidity energy:

$$E_{internal}(s,t) = \left(\frac{\alpha(s,t)}{2} \left| v_s(s,t) \right|^2 + \frac{\beta(s,t)}{2} \left| v_{ss}(s,t) \right|^2 \right)$$

where α controls the tension of the contour, while β controls its rigidity. Thus, this functions determinate how the snake can stretch or bend at any point s of the spline curve. The second terms couples the snake to the image:

$$E_{ext\_potential}(s,t) = P(v(s,t))$$

where $P(v(s,t))$ denotes a scalar potential function defined on the image plane. It is responsible for attracting the contour towards the object in the image (external energy). Therefore, it can be expressed as a weighted combination of energy function.

To apply snakes to images, external potentials are designed whose local minima coincides with intensity *extrema*, edges and other image features of interest. For example, the contour will be attracted to intensity edges in an image by choosing a potential

$$P(v(s,t)) = -c \left| \nabla \left[G_\sigma * I(x, y) \right] \right.$$

where c controls the magnitude of the potential, ∇ is the gradient operator and $G_\sigma * I(x,y)$, denotes the image convolved with a Gaussian smoothing filter.

In accordance with the calculus of variations, the contour $v(s,t)$ that minimizes the energy of (1) must satisfy the Euler-Lagrange equation. Moreover, the Lagrange equation of motion for a snake with the internal and external energy given by equation (1) is:

$$\mu \frac{\partial^2 v}{\partial t^2} + \gamma \frac{\partial v}{\partial t} - \frac{\partial}{\partial s} \left(\frac{\alpha(s)}{2} \left| v_s(s,t) \right|^2 \right) +$$

$$\frac{\partial^2}{\partial s^2} \left(\frac{\beta(s)}{2} \left| v_{ss}(s) \right|^2 \right) + \nabla P(v(s,t)) = 0$$

with a mass density μ and a damping density *γ*. This leads to dynamic deformable models that unify the description of shape and motion, making it possible to quantify not just static shape, but also shape evolution through time. The first two terms of this partial differential equation represent inertial and damping forces. The remaining terms represent the internal stretching, the bending forces and the external forces. Equilibrium is achieved when the internal and external forces balance and the contour comes to rest, i.e., inertial and damping forces are zero, which yields the equilibrium condition.

Traditional snake models are known to be limited in several aspects, such as their sensitivity to the initial contours. These are non-free parameters and do not handle changes in the topology of the shape. That is, when considering more than one object in the image, for instance for an initial prediction of $v(s,t)$ surrounding all of them, it is not possible to detect all the objects. Special topology-handling procedures must be added. Some techniques have been proposed to solve these drawbacks. These techniques are based on information

fusion, dealing with ACM in addition to curvature driven flows and geometrical distance conditions (Solaiman, 1999), (Caselles, 1997).

GEODESIC ACTIVE CONTOUR MODEL

The geodesic active contour model is based on the computation of a minimal distances curve. Thereby, the AC evolves following the geometric heat flow equation. Let us consider a particular class of snake models in which the rigidity coefficient is set to zero, that is $\beta = 0$. Two main reasons motivate this selection: i) this will allow us to derive the relation between these energy-based active contours and geometric curve evolution ones, ii) the regularization effect on the geodesic active contours comes from curvature-based curve flows, obtained only from the other terms in equation (1). This will allow to achieve smooth curves in the proposed approach without having the high-order smoothness given by $\beta \neq 0$ in energy-based approaches.

Moreover, this smoothness component in equation (1) appears in order to minimize the total squared curvature. It is possible to prove that the curvature flow used in the geodesic model decreases the total curvature. The use of the curvature-driven curve motions as smoothing terms was proven to be very efficient. Therefore, curve smoothing will be obtained also with $\beta = 0$, having only the first regularization term. Assuming this, equation (1) may be reduced to:

$$E_{geo}(s,t) = \int_0^1 \left(\frac{\alpha(s,t)}{2} |v_s(s,t)|^2 \right) ds - \int_0^1 c |\nabla I [v(s,t)]| ds$$

(2)

Observe that, by minimizing the functional of equation (2), we are trying to locate the curve at the points of maxima $|\nabla I|$ (acting as edge detector) while keeping a certain smoothness in the curve (object boundary). This is actually the goal in the general formulation (equation 1) as well.

It is possible to extend equation (2), generalizing the edge detector part in the following way: let $\{g: [0, \infty[\rightarrow \mathcal{R}^+\}$ be a strictly decreasing function, which acts as a function of the image gradient used for the stopping criterion. Hence we can replace $-|\nabla I|$ with $g(|\nabla I|)^2$, obtaining a general energy function:

$$E_{geo}(s,t) = \int_0^1 \left(\frac{\alpha(s,t)}{2} |v_s(s,t)|^2 \right) ds - \int_0^1 cg(|\nabla I [v(s,t)])^2 ds$$

(3)

The solution of the particular energy snake model of equation (3) is given by a geodesic curve in a Riemann space induced from the image $I(x,y)$, where a geodesic curve is a local minimal distance path between given points. To show this, the classical Maupertuis' principle from dynamical systems together with the Fermat's principle is used.

By assuming $E_{internal} = E_{ext\_potential}$, it is possible to reduce the minimization of equation (1) to the following form:

$$\min_{v(s)} = \int_0^1 g(|\nabla I(x, y)|(v(s)) \cdot |v_s(s)| ds$$

This is done by using Euler-Lagrange, and defining an embedding function of the set of curves $v(s)$, $\psi(s, t)$. That is an implicit representation of $v(s)$, assuming that $v(s)$ is a level set of a function $\psi(s,t){:}[0,a] *[0,b] \rightarrow \mathcal{R}$, the following equation for curve/surface evolution is derived:

$$\frac{\partial \psi}{\partial t} = g(\nabla I)(C + K)|\nabla \psi|$$

where C is a positive real constant and K is the Euclidean curvature in the direction of the normal.

To summarize the force $(C+K)$ acts as the internal force in the classical energy-based snake model, smoothness being provided by the curvature part of the flow. The heat-flow K ψ is the regularization curvature flow that replaces the second order smoothness term in equation (1). The external-image dependent force is given by the stopping function, $g(\nabla I)$. Thus this function stops the evolving curve when it arrives at the object's boundaries.

The advantage of using a level set is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects (this is called the *Eulerian approach)*. Moreover, level set representation can handle topological shape changes as the surface evolves and it is less sensitive to initialisation. Figure 1 shows

E

the results of the level set approach applied to a bio-medical colour image. Figure 1.a) shows the original image, b) show the initialization, which is composed by multiple curves. This is also an advantage over the classical ACM. Figure 1.c) shows the results after 50 iterations and d) show the final ROI detection.

Figure 2 shows different results for ROI detection in **biomedical images** with the geodesic active contour model. Column (a) and (c) show the original image and (b) and (d) the final segmentation result.

The model is efficient in time and accuracy. However, there are also some drawbacks in terms of effi-ciency and convergence. It has non-free parameters, ψ is dependent of the time step, Δt, and the spatial one.

DEFORMABLE FINITE ELEMENT MODEL

A powerful approach to computing the local minima of a functional such as equation (1) is to construct a dynamical system that is governed by the energy minimization function and allows the system to evolve to equilibrium.

Figure 1. Geodesic active contour model with multiple initializations

(a)   *(b)*   *(c)*   *(d)*
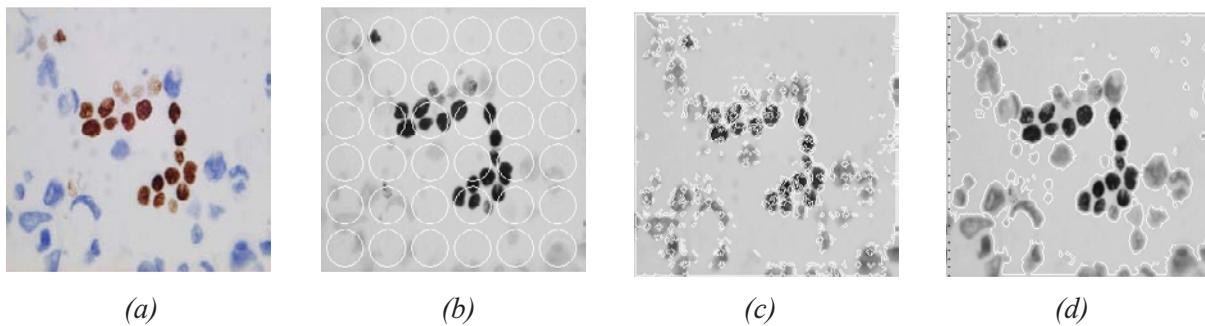
Figure 2. Results of ROI detection with the geodesic active contour model

(a)   *(b)*   *(c)*   *(d)*
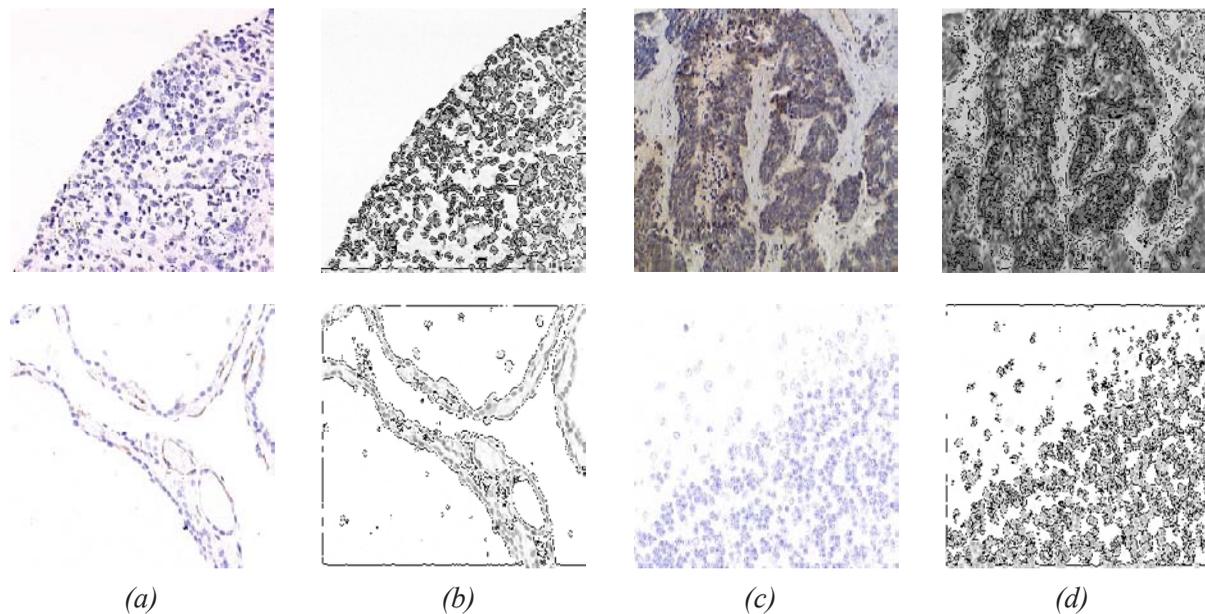
In order to compute a minimum energy solution numerically, it is necessary to discretize the energy (1). Thus, the continuous model, *v* is represented in discrete form by a vector *U* of shape parameters associated with the local-support basis functions. The discretized version of the Lagrangian dynamics equation may be written as a set of second-order ordinary differential equations for the discrete nodal points displacements *U*, that is a vector of the *(Δx, Δy, Δz)* displacements of the n nodal points that represents the ROI, thus:

$$M\ddot{U} + C\dot{U} + KU = \vec{F} \qquad (4)$$

That is the governing equation of the physical model, which is characterized by its mass matrix *M*, its stiffness matrix *K* and its dumping matrix *C* and the vector on the right hand side is describing the *x, y,* and *z* components of the forces acting on the nodes. Equation (4) is known as the governing equation in the FEM method, and may be interpreted as assigning a certain mass to each nodal point and a certain material stiffness and damping between nodal points.

The main drawback of the FEM is the large computational expense. Another important problem when using the FEM for vision is that all the degrees of freedom are coupled. Therefore, closed-form solutions are not possible. In order to solve the problem the system may be diagonalized by means of a linear transform into the vibration modal space of the mesh modelling the ROI (Pentland, 1991). Thus, vector *U* is transformed by a matrix *P* derived from the free vibrations modes of the equilibrium equation. Therefore, an eigenvalue problem can be derived with the basis set of eigensolutions composed by $\{w_i, \phi_i\}$. The eigenvector ϕ_i is called

the ith mode's shape vector and w_i is the corresponding frequency of vibration. Each eigenvector ϕ_i consists of the *(x, y, z)* displacement for each mode that parameterize the ROI. Usually the basis set is reduced by its Karhunen-Loeve (KL) expansion.

The model maybe also extended to contain a set of ROIs. Thus, for a given image in a training set, a vector containing the largest vibration modes describing the different deformable ROI surfaces is created. This random vector may be statistically constrained by retaining the most significant variation modes of its KL expansion on the image data set. By these means, the conjunction of ROI surfaces may be deformed according to the variability observed in the training set.

Figure 3 shows the results of the FEM model applied to a brain **magnetic resonance** image and a lung computed **tomography** image. Figure 3.a) shows the original structure inside the initial mesh surface and b) show the model obtained after evolve the spherical mesh by means of equation (3). Figure 3. c) shows the original lung structure and d) the final model when using also a spherical mesh as initial surface.

FUTURE TRENDS

Deformable models are suitable for different applications and domains such as computer vision, computational fluid dynamics, computer graphics and biomechanics. However, each model usually is application dependent and further improvements and processing of the model should be done to achieve satisfactory results. Techniques based on region properties, fuzzy logic theory and combination of different models have

Figure 3. Results of the FEM model

(a)   *(b)*   *(c)*   *(d)*
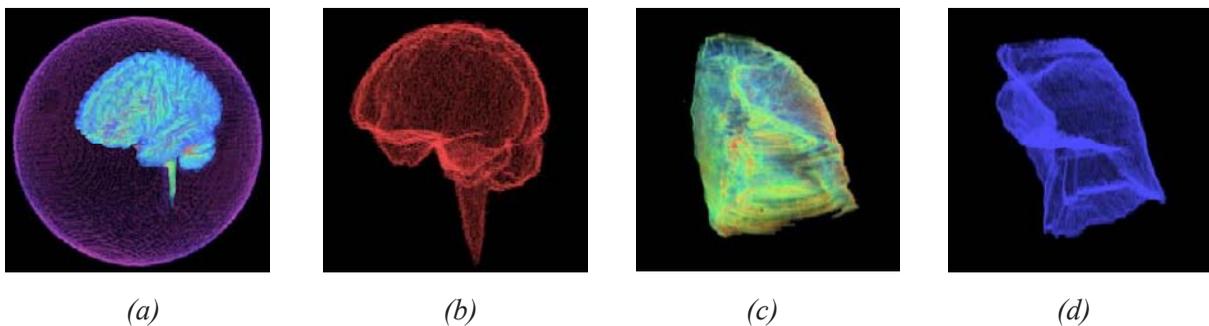
already been suggested (Ray, 2001), (Bueno, 2004), (Yu, 2002), (Bridson, 2005).

Moreover, these techniques have been broadly used to model real life phenomena, like fire, water, cloth, fracturing materials, etc. Results and models from these research areas may be of practical significance if they are applied in Artificial Vision.

CONCLUSION

Energy minimizing active models has been shown to be a powerful technique to detect, track and model interfaces and shapes of ROIs. Since the work on ACM by (Kass, 1987) energy minimizing active models have been applied and improved by using different mathematical and physical techniques. The most promising models are those presented here: the geodesic and the FEM one for ROI tracking and modeling respectively.

The models are suitable for color and 3D images. The geodesic model may handle topological changes on the ROI surface and in not sensitive to initialization. The FEM model may represent the relative location of different ROI surfaces and it is able to accommodate their significant variability across different images of the ROI. The surfaces of each ROI are parameterized by the amplitudes of the vibration modes of a deformable geometrical mesh, which can handle small rotations and translations. However, as mentioned before there is still room for further improvements within these models.

REFERENCES

Bridson, R. Teran, J. Molino, N. and Fedkiw, R. (2005). Adaptative Physics Based Tetrahedral Mesh Generation Using Level Sets, *Engineering with Computers* 21, 2-18.

Bueno G., Martínez A. (2004). Fuzzy-Snake Segmentation of Anatomical Structures Applied to CT Images. *Lecture Notes in Computer Science*, vol. 3212, pp. 33-42.

Caselles V., Kimmel R., and Sapiro G. (1997). Geodesic Active Contours, *Int. J. Computer Vision,* vol. 22(1), pp. 61-79.

Cootes T.F., Cooper D., Taylor C. and Graham J. (1992). A Trainable Method of Parametric Shape Description. *Image and Vision Computing*, vol. 10, pp.289-294.

Duncan J.S. and Ayache N. (2000). Medical Image Analysis: Progress over Two Decades and the Challenges Ahead. *IEEE Trans. on PAMI*, vol. 22, pp. 85-106.

Ivins J., Porrill J. (1994). Active Region Models for Segmenting Medical Images. *IEEE Trans. on Image Processing*, pp. 227-231.

Kass M., Witkin A., Terzopoulos D. (1987). Snakes: Active Contour Models. *Int. J. of Computer Vision*, pp. 321–331.

Malladi R., Sethian J. A. and Vemuri B.C. (1995). Shape Modelling with Front Propagation: A Level Set Approach. *IEEE Trans. on PAMI*, vol. 17, pp. 158-175.

McInerney T. and Terzopoulos D. (1996). Deformable models in medical image analysis: A survey. *Medical Image Analysis,* vol. 2(1), pp. 91-108.

Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computation Physics*, vol. 79, pp. 12–49.

Pentland A. and Sclaro. S. (1991). Closed-form solutions for physically based shape modelling and recognition. *IEEE Trans. Pattern Anal. Machine Intelligence*, vol. 13, pp. 730-742.

Ray N., Havlicek J., Acton S. T., Pattichis M. (2001). Active Contour Segmentation Guided by AM-FM Dominant Component Analysis. *IEEE Int. Conference on Image Processing* pp. 78-81.

Solaiman B., Debon R. Pipelier F., Cauvin J.-M., and Roux C. (1999). Information Fusion: Application to Data and Model Fusion for Ultrasound Image Segmentation. *IEEE Trans. on BioMedical Engineering*, vol. 46(10), pp. 1171-1175.

Terzopoulos D. and Metaxas D. (1991) Dynamic 3D Models with Local and Global Deformations:Deformable Superquadrics. *IEEE Transactions on PAMI*, vol. 13(7), pp. 703 – 714.

Wang H., Ghosh B. (2000) Geometric Active Deformable Models in Shape Modelling. *IEEE Transactions on Image Processing*, vol. 9(2), pp. 302-308.

Yu Z. and Bajaj C. (2002). Image Segmentation Using Gradient Vector Diffusion and Region Merging.

IEEE Int. Conference on Pattern Recognition, vol. 2, pp. 20941.

KEY TERMS

Active Model: It is a numerical technique for tracking interfaces and shapes based on partial differential equations. The model is a curve or surface which iteratively deforms to fit to an object in an image.

Conventional Mathematical Modeling: The applied science of creating computerized models. That is a theoretical construct that represents a system composed by set of region of interest, with a set of parameters, both variables together with logical and quantitative relationships between them, by means of mathematical language to describe the behavior of the system. Parameters are determined by finding a curve in 2D or a surface in 3D, each patch of which is defined by a net of curves in two parametric directions, which matches a series of data points and possibly other constraints.

Finite Element Method: Numerical technique used for finding approximate solution of partial differential equations (PDE) as well as of integral equations. The solution approach is based either on eliminating the differential equation completely (steady state problems), or rendering the PDE into an equivalent ordinary differential equation, which is then solved using standard techniques such as finite differences.

Geodesic Curve: In presence of a metric, geodesics are defined to be (locally) the shortest path between points on the space. In the presence of an affine connection, geodesics are defined to be curves whose tangent vectors remain parallel if they are transported along it. Geodesics describe the motion of point particles.

Karhunen-Loeve: Mathematical techniques equivalent to Principal Component Analysis transform aiming to reduce multidimensional data sets to lower dimensions for analysis of their variance.

Modal Analysis: Study of the dynamic properties and response of structures and or fluids under vibrational excitation. Typical excitation signals can be classed as impulse, broadband, swept sine, chirp, and possibly others. The resulting response will show one or more resonances, whose characteristic mass, frequency and damping can be estimated from the measurements.

Tracking: Tracking is the process of locating a moving object (or several ones) in time. An algorithm analyses the image sequence and outputs the location of moving targets within the image. There are two major components of a visual tracking system; *Target Representation and Localization* and *Filtering and Data Association*. The 1st one is mostly a bottom-up process which involve segmentation and matching. The 2nd one is mostly a top-down process, which involves incorporating prior information about the scene or object, dealing with object dynamics, and evaluation of different hypotheses.

Ensemble of ANN for Traffic Sign Recognition

M. Paz Sesmero Lorente
Universidad Carlos III de Madrid, Spain

Juan Manuel Alonso-Weber
Universidad Carlos III de Madrid, Spain

Germán Gutiérrez Sánchez
Universidad Carlos III de Madrid, Spain

Agapito Ledezma Espino
Universidad Carlos III de Madrid, Spain

Araceli Sanchis de Miguel
Universidad Carlos III de Madrid, Spain

INTRODUCTION

"Machine Learning (ML) is the subfield of Artificial Intelligence conceived with the bold objective to develop computational methods that would implement various forms of learning, in particular mechanisms capable of inducing knowledge form examples or data" (Kubat, Bratko & Michalski, 1998, p. 3).

The simplest and best-understood ML task is known as supervised learning. In supervised learning, each example consists of a vector of features (**x**) and a class (y). The goal of the learning algorithm is, given a set of examples and their classes, find a function, *f*, that can be applied to assign the correct class to new examples. When the function *f* takes values from a discrete set of classes $\{C_1, \ldots, C_K,\}$, *f* is called a *classifier* (Dietterich, 2002).

In the last decades it has been proved that learning tasks in which the unknown function *f* takes more than two values (multi-class learning problems) the better approach is to decompose the problem into multiple two-class classification problems (Ou & Murphey, 2007) (Dietterich, & Bakiri, 1995) (Massulli & Valentini, 2000).

This article describes the implementation of a system whose main task is to classify prohibition road signs into several categories. In order to reduce the learning problem complexity and to improve the classification performance, the system is composed by a collection (ensemble) of independent binary classifiers. In the proposed approach, each binary classifier is a single-output neural network (NN) trained to distinguish a particular road sign kind from the others.

The proposed system is a part of a Driver Support System (DSS) supported by the Spanish Government under project TRA2004-07441-C03-C02. For this reason, one of the main system requirements is that it should be implemented in hardware in order to use it aboard a vehicle for real time categorization. In order to fulfill this constraint, a reduction in the number of features that describe the instances must be performed. As consequence if we have *k* generic road sign types we will use *k* binary NN and *k* feature selection process will be executed.

BACKGROUND

It is known that road signs carry essential information for safe driving. Among other things, they permit or prohibit certain maneuvers, warn about risk factors, set speed limits and provide information about directions, destinations, etc. Therefore, road sign recognition is an essential task for the development of an autonomous Driver Support System.

In spite of the increasing interest in the last years, traffic sign recognition is one of the less studied subjects in the field of Intelligent Transport Systems. Approaches in this area have been mainly focused on the resolution of other problems, such as road border detection (Dickmanns & Zapp, 1986) (Pomerlau & Jochem, 1996) or the recognition of obstacles in the

vehicle's path such as pedestrians (Franke, Gavrilla, Görxig, Lindner, Paetzold & Wöhler, 1998) (Handmann, Kalinke, Tzomakas, Werner & Seelen, 1999) or other vehicles (Bertozzy & Broggi, 1998).

When the number of road sign types is large, road sign recognition task is separated in two processes: detection and classification. Detection process is responsible for the localization and extraction of the potential signs from images captured by cameras. Only when the potential signs have been detected they can be classified as one of the available road sign-types.

In the published researches, detection is based on color and/or shape of traffic signs (Lalonde & Li, 1995). On the other hand, to solve the classification task several ML algorithms have been used. Among the used techniques it is worth mentioning: The Markov Model (Hsien & Chen, 2003), Artificial Neural Networks (Escalera, Moreno, Salich & Armingol, 1997) (Yang, Liu & Huang, 2003), Ring Partitioned Method (Soetedjo & Yammada 2005), the Matching Pursuit Filter (Hsu & Huang 2001) or the Laplace Kernel classifier (Paclík, Novovicová, Pudil, & Somol, 1999).

A NEURAL NETWORK BASED SYSTEM FOR TRAFFIC SIGN RECOGNITION

In this work, we present the architecture of a system whose task is to classify prohibition road signs into several categories. This task can be described as a supervised learning problem in which the input information comes from a set of road signs arranged in a fixed number of categories (classes) and the goal is to extract, from the input data, the real knowledge needed to classify correctly new signs.

The proposed system is a Multilayer Perceptron (MLP) based classifier trained with the Back-Propagation algorithm. In order to integrate this classification system into a DSS capable to perform real-time traffic sign categorization, a hardware implementation on Field Programmable Gate Array (FPGA) is necessary.

With the aim of reducing the problem complexity, an ensemble of specialized neural networks is proposed. In addition and due to the strict size limitations of ANN implementation on FPGAs (Zhu & Sutton, 2003) the construction of each specialized MLP is combined with a specific reduction in the number of features that describes the examples.

Traffic Sign Pre-Processing

Since the signs to be classified are embodied in images acquired by a camera attached to a moving vehicle, it can be assumed that the signs have a varying size (signs get bigger as the vehicle moves toward them). Therefore, once the traffic signs have been detected, the first step is to normalize them to a specific size. The aim of this process is to ensure that all the signs (examples) are described by the same number of pixels (features). In our approach we have used 32x32 pixel signs.

Once the signs have been normalized, a grayscale conversion is performed. Since the original images are represented in the RGB (Red, Green and Blue) color space, this conversion is done by adding the red, green and blue values for each pixel and dividing by three. As result of both processes, each road sign is transformed into a 1024 element vector in which each pixel is represented by a real number in the range [0.0, 1.0].

System Architecture

The general framework of the proposed system (Figure 1) is composed of two modules: the *Data Preprocessing Module* (DPM) and the *Classification Module* (CLM). The DPMs function is to select from among the 1024 attributes that describe a sign the subset that each specialized neural network inside the CLM must receive. On the other hand, the CLMs function is to classify each input data set as one of the available prohibition road sign-types. Since this module is composed of several independent classifiers, in order to obtain the final classification, an integration of the individual predictions is required.

To build both, the DPM and the CLM, a new data encoding schema is necessary. In particular, the multiclass problem has to be decomposed into a set of binary subproblems.

Data Preprocessing Module

Practical experience shows that using as much as possible input information (features) does not imply higher output accuracy. Feature subset selection (Witten & Frank, 2005) (Hall, 1998) is the procedure of selecting just the relevant information, avoiding irrelevant and redundant information and reducing the learning task dimensionality.

E

Figure 1. General framework of the proposed architecture
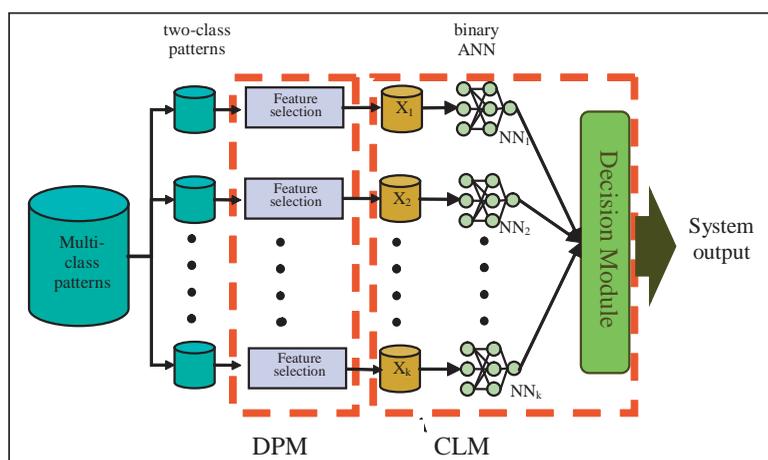
The proposed architecture adopts a model in which the feature subset that describes an example is not unique but depends on the task associated to each classifier. In other words, since the classification problem is divided in *k* binary sub-problems, *k* feature selection procedures are necessary.

In this work, the feature selection module has been built using the Weka tool (Witten & Frank, 2005). At first, several feature selection algorithms from those included in Weka were considered (Sesmero, Alonso-Weber, Gutiérrez, Ledezma & Sanchis, 2007). After analyzing both, the feature set size and the experimental results, combination of Best First (Russell & Norvig, 2003) and Correlation-based Feature Selection (Hall, 1998) was selected as base for the DPM construction.

Classification Module

The Classification Module is based on an One Against All (OAA) model. In this modeling, the final classifier is composed of a collection of binary classifiers where each of them is specialized in discriminating a specific road sign type from the others. Therefore, for a classification problem where *k* road sign types have to be separated, this approach results in a system in which for each existing class a different NN is used.

Decomposing the global classifier into a set of independent NN not only reduces the complexity problem but also permits that the DPM is able to select the most significant attribute set for each binary classification

task. In addition, each NN can have its own architecture (number of hidden nodes, activation function, learning rate, etc) and since there is no connection between the individual networks, the training can be performed distributing the work on several processors.

ANNs Output Combination

Once the binary NN's have been trained the global classifier system can be generated. However, since each classifier makes its own prediction, a decision module that integrates the results from the set of classifiers and produces a unique final classification is required. Experimentally it is found that, for the proposed classification task, the most efficient decision criterion is selecting the NN with the highest output value. Therefore, the formula used in the decision module is:

$$f(\overline{x}, f_1, f_2, ... f_k) = \arg \max_{i=1,...,k}(f_i) \tag{1}$$

where f_i is the output value of the neural network associated to the i-th class.

Classification Process

When the system receives an unlabeled road sign to be classified in some of the fixed categories, such sign is sent to each classifier's input module. The DPM selects the pixel subset according to its relevant attribute list. The chosen pixels are used as the input for the asso-

E

ciated ANN, which applies its knowledge to make a prediction. The individual predictions are sent to the decision module that carries out an integration of the received information and produces a unique final classification. This process is shown is Figure 2.

Empirical Evaluation

The proposed system has been validated over 5000 examples arranged in ten generic kinds of prohibition road signs: no pedestrians, no left/right turn ahead, no stopping and no parking, no overtaking, and 20-30-40-50-60 and 100 km speed limits.

In order to evaluate our approach, three classification methods have been compared:

- The direct multi-class approach,
- The OAA approach with the full feature space and,
- The OAA approach with feature selection.

In the direct multi-class approach (experiment 1), the classification problem has been solved with a MLP with 1024 (32×32) input nodes, one hidden layer with 50 neurons and one output layer with 10 neurons. In this approach the class associated to each learning pattern is encoded using a vector **C**, which has as many components c_i as existing class (10). The component value c_i will be 1 if the sign belongs to class i, and 0 in any other case.

In the OAA approach with the full feature space (experiment 2) the previous net is split into ten binary ANN. In other words, this approach uses 10 binary

MLP with 1024 input nodes, 36 hidden nodes and 1 output node.

Finally, in the OAA approach with Feature Selection (experiment 3), the problem is solved with an ensemble containing 10 binary MLP with 36 hidden nodes in each. The number of input units and, therefore, the feature space used by each ANN is determined by the DPM. This number is shown in Table1.

In order to build the binary classifiers used in the last two experiments, a new class encoding schema is necessary. In both cases, the class associated with each patter is encoded using a bit. Since in both experiments, the i-th binary classifier is trained to distinguish the class *i* from all the other class, the new encoding is equivalent to select the c_i component from the previous codification.

In Table 2 we show the estimate classification accuracy for the described experiments when a 10-fold cross validation process is used.

The experimental evaluation reflects that splitting the classification task into binary subtasks (experiment 2) increases the classification accuracy.

On the other hand, the loss of classification accuracy when the feature selection process is performed (experiment 3) is not very significant compared with the benefits of the drastic input data reduction.

FUTURE TRENDS

The future work will be mainly focused on extending the system in order to cope with regulatory, warning, indication, etc, signs, i.e, with a bigger number of classes. This task will allow us to investigate and de-

Figure 2. Classification process
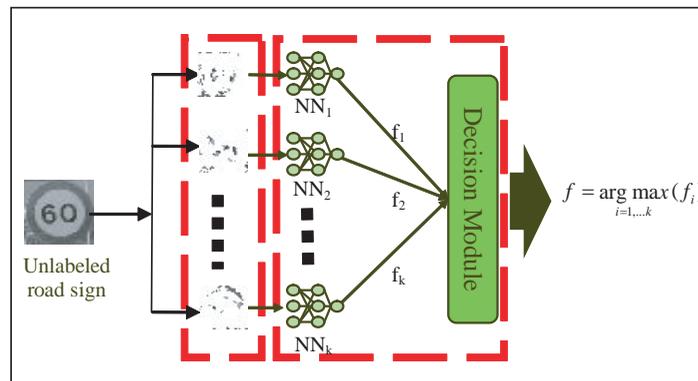
Table 1. Number of selected features; in the first column appears the label of each class

Class	Prohibition road sign	Number of selected features
C1	no pedestrians	116
C2	no (left, right) turn ahead	91
C3	stopping and no parking	44
C4	no passing	114
C5	60 km speed limit	114
C6	50 km speed limit	110
C7	40 km speed limit	100
C8	30 km speed limit	114
C9	20 km speed limit	103
C10	100 km speed limit	87

Table 2. Summary of estimate classification accuracy (percentage)

Experiment	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Global
1	100	98,0	100	100	96,0	100	90,0	100	96,0	100	97,8
2	100	100	100	100	96,0	100	96,0	100	98,0	100	99,0
3	100	100	98,0	100	92,0	96,0	88,0	100	98,0	98,0	97,0

velop new procedures that will contribute to the design of a more versatile system.

In the design of this new system, other multi-class approach such as the One Against Higher Order Modeling (Lu & Ito, 1999) and the Error-Correction Output Code (Dietterich & Bakiri, 1995) would be analyzed.

CONCLUSION

In this work, an architecture for traffic sign classification has been described. The software implementation shows very high recognition rates. For this reason this architecture can be considered as a good solution for the traffic sign classification problem.

Moreover, the features of this architecture make it possible to implement this system on FPGAs and, therefore, to use it in real-time applications.

REFERENCES

Bertozzi, M. & Broggi, A. (1998). GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Transactions on Image Processing,* (7), 1, 62-81.

Dickmanns, E.D. & Zapp, A. (1986). A Curvature-Based Scheme for Improving Road Vehicle Guidance by Computer Vision. *Proceedings of the SPIE Conference on Mobile Robots,* 161-168.

Dietterich, T.G. (2002). Ensemble Learning. The Handbook of Brain Theory and Neural Networks, Second edition, (Arbib, M.A. Editors). Cambridge, MA: The MIT Press.

Dietterich, T.G. & Bakiri, G. (1995). Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research* (2), 263-286.

Escalera, A., Moreno, L.E., Salich, M.A. & Armingol, J.M. (1997). Road Traffic Sign Detection and Classi-

fication. *IEEE Transactions on Industrial Electronics,* (44) 6, 848-859.

Franke, U., Gavrila D., Görxig, S., Lindner, F., Paetzold, F. & Wöhler, C. (1998). Autonomous Driving Goes Downtown. *IEEE Intelligent Systems,* (13) 6, 40-48.

Hall, M.A. *Correlation-based Feature Selection for Machine Learning*. Ph.D diss. Hamilton, NZ: Waikato University Department of Computer Science (1998)

Handmann, U., Kalinke, T., Tzomakas, C., Werner, M. & Seelen, W. (1999). An Image Processing System for Driver Assistance. *Image and Vision Computing,* (18) 5, 367-376.

Hsien, J.C. & Chen, S.Y. (2003). Road Sign Detection and Recognition Using Markov Model. *14th Workshop on Object-Orient Technology and Applications*, 529-536.

Hsu S.H. & Huang, C.L. (2001). Road sign detection and recognition using matching pursuit method. *Image and Vision Computing*, (19) 3, 119-129

Kubat, M., Bratko, I. & Michalski, R. (1998). A Review of Machine Learning Methods. *Machine Learning and Data Mining: Methods and Applications*, Michalski, R.S., Bratko, I., and Kubat, M. (Editors), John Wiley and Sons, 3-70.

Lalond, M. & Li, Y. (1995). Road Sign Recognition. *Collection scientifique et technique*, CRIM-IIT-95/09-35.

Lu, B.L. & Ito, M. (1999). Task Decomposition and Module Combination Based on Class Relations: A Modular Neural Network for Pattern Classification. IEEE Transaction on Neural Networks, (10) 5, 1244-1256.

Masulli, F. & Valentini, G. (2000). Comparing Decomposition Methods for Classification. *4th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, 788-791.

Ou, G. & Murphey Y. L. (2007). Multi-Class Pattern Classification using neural Networks. *In Pattern Recognition.* (40) 1, 4-18.

Paclík, P. Novovicová, J., Pudil, P. & Somol, P. (1999). Road sign classification using the Laplace Kernel Classifier. *Proceedings of the 11th Scandinavian Conference on Image Analysis*, (1), 275-282.

Pomerleau D. & Jochem, T. (1996). Rapidly Adapting Machine Vision for Automated Vehicle Steering. *IEEE Expert Magazine,* (11) 2, 19-27.

Russell, S.J. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Sesmero, M.P., Alonso-Weber, J.M., Gutiérrez, G., Ledezma, A. & Sanchis A. (2007). Testing Feature Selection in Traffic Signs. *Proceedings of 11th International Conference on Computer Aided Systems Theory,* 396-398.

Soetedjo, A. & Yamada, K. (2005). Traffic Sign Classification Using Ring Partitioned Method. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences archive* (E88-A) 9, 2419-2426.

Witten, I.H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Yang, H.M., Liu, C.L. & Huang, S.M. (2003). Traffic Sign Recognition in disturbing Environments. *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems*, 28-31.

Zhu, J. & Sutton, P. (2003). FPGA Implementations of Neural Networks – A survey of a Decade of Progress. *In Field-Programmable Logic and Applications. Lecture Notes in Computer Science*, (2778), 1062-1066.

KEY TERMS

Artificial Neural Network: Structure composes of a group of interconnected artificial neurons or units. The objective of a NN is to transform the inputs into meaningful outputs.

Correlation-based Feature Selection: Feature Selection[*] algorithm which heuristic measures the correlation between attributes and rewards those feature subsets in which each feature is highly correlated with the class and uncorrelated with other subset features.

[*]**Feature Selection:** Process, commonly used in machine learning, of identifying and removing as much of the irrelevant and redundant information as possible.

Feature Space: n-dimensional space where each example (pattern) is represented as a point. The dimension of this space is equal to the number of features used to describe the patterns.

Field Programmable Array (FPGA): A FPGA is an integrated circuit that can be programmed in the field after manufacture.

K-Cross-Validation: Method to estimate the accuracy of a classifier system. In this approach, the dataset, D, is randomly split into K mutually exclusive subsets (folds) of equal size (D_1, D_2, …, D_k) and K classifiers are built. The i-th classifier is trained on the union of all D_j / j≠i and tested on D_i. The estimate accuracy is the overall number of correct classifications divided by the number of instances in the dataset.

Machine Learning: Computer Scientific field focused on the design, analysis and implementation, of algorithms that learn from experience.

One Against All: Approach to solve multi-class classification problems which creates one binary problem for each of the K classes. The classifier for class *i* is trained to distinguish examples in class *i* from all other examples.

Weka: Collection of machine learning algorithms for solving data mining problems implemented in Java and open sourced under the GPL.

Ensemble of SVM Classifiers for Spam Filtering

Ángela Blanco
Universidad Pontificia de Salamanca, Spain

Manuel Martín-Merino
Universidad Pontificia de Salamanca, Spain

INTRODUCTION

Unsolicited commercial email also known as Spam is becoming a serious problem for Internet users and providers (Fawcett, 2003). Several researchers have applied machine learning techniques in order to improve the detection of spam messages. Naive Bayes models are the most popular (Androutsopoulos, 2000) but other authors have applied Support Vector Machines (SVM) (Drucker, 1999), boosting and decision trees (Carreras, 2001) with remarkable results. SVM has revealed particularly attractive in this application because it is robust against noise and is able to handle a large number of features (Vapnik, 1998).

Errors in anti-spam email filtering are strongly asymmetric. Thus, false positive errors or valid messages that are blocked, are prohibitively expensive. Several authors have proposed new versions of the original SVM algorithm that help to reduce the false positive errors (Kolz, 2001, Valentini, 2004 & Kittler, 1998). In particular, it has been suggested that combining non-optimal classifiers can help to reduce particularly the variance of the predictor (Valentini, 2004 & Kittler, 1998) and consequently the misclassification errors. In order to achieve this goal, different versions of the classifier are usually built by sampling the patterns or the features (Breiman, 1996). However, in our application it is expected that the aggregation of strong classifiers will help to reduce more the false positive errors (Provost, 2001 & Hershop, 2005).

In this paper, we address the problem of reducing the false positive errors by combining classifiers based on multiple dissimilarities. To this aim, a diversity of classifiers is built considering dissimilarities that reflect different features of the data.

The dissimilarities are first embedded into an Euclidean space where a SVM is adjusted for each measure. Next, the classifiers are aggregated using a voting strategy (Kittler, 1998). The method proposed has been applied to the Spam UCI machine learning database (Hastie, 2001) with remarkable results.

THE PROBLEM OF DISSIMILARITIES REVISITED

An important step in the design of a classifier is the choice of the proper dissimilarity that reflects the proximities among the objects. However, the choice of a good dissimilarity for the problem at hand is not an easy task. Each measure reflects different features of the dataset and no dissimilarity outperforms the others in a wide range of problems. In this section, we comment shortly the main differences among several dissimilarities that can be applied to model the proximities among emails. For a deeper description and definitions see for instance (Cox, 2001).

The Euclidean distance evaluates if the features that codify the spam differ significantly among the messages. This measure is sensible to the size of the emails. The cosine dissimilarity reflects the angle between the spam messages. The value is independent of the message length. It differs significantly from the Euclidean distance when the data is not normalized. The correlation measure checks if the features that codify the spam change in the same way in different emails. Correlation based measures tend to group together samples whose features are linearly related. The correlation differs significantly from the cosine if the mean of the vectors that represents the emails are not zero. This measure is distorted by outliers. The Spearman rank correlation avoids this problem by computing a correlation between the ranks of the features. Another kind of correlation measure that helps to overcome the problem of outliers is the kendall-τ index which is related to the Mutual Information probabilistic measure.

When the emails are codified in high dimensional and noisy spaces, the dissimilarities mentioned above are affected by the `curse of dimensionality' (Aggarwal, 2001 & Martín-Merino, 2004). Hence, most of the dissimilarities become almost constant and the differences among dissimilarities are lost (Hinneburg, 2000 & Martín-Merino, 2005). This problem can be avoided selecting a small number of features before the dissimilarities are computed.

COMBINING DISSIMILARITY BASED CLASSIFIERS

In this section, we explain how the SVM can be extended to work directly from a dissimilarity measure. Next, the ensemble of classifiers based on multiple dissimilarities is presented. Finally we comment briefly the related work.

The SVM is a powerful machine learning technique that is able to deal with high dimensional and noisy data (Vapnik, 1998). In spite of this, the original SVM algorithm is not able to work directly from a dissimilarity matrix. To overcome this problem, we follow the approach of (Pekalska, 2001). First, the dissimilarities are embedded into an Euclidean space such that the inter-pattern distances reflect approximately the original dissimilarity matrix. Next, the test points are embedded via a linear algebra operation and finally the SVM is trained and evaluated. We comment briefly the mathematical details.

Let $D \in R^{nxn}$ be the dissimilarity matrix made up of the object proximities for the training set. A configuration in a low dimensional Euclidean space can be found via a metric multidimensional scaling algorithm (MDS) (Cox, 2001) such that the original dissimilarities are approximately preserved. Let $\mathbf{X} = [x_1,\ldots,x_n]^T \in R^{nxp}$ be the matrix of the object coordinates for the training patterns. Define $\mathbf{B} = \mathbf{X}\,\mathbf{X}^T$ as the matrix of inner products which is related to the dissimilarity matrix via the following equation:

$$\mathbf{B} = -1/2 \, \mathbf{J} \, \mathbf{D}^{(2)} \, \mathbf{J} \tag{1}$$

where $\mathbf{J} = \mathbf{I} - 1/n \, \mathbf{1}\,\mathbf{1}^T \in R^{nxn}$ is the centering matrix, $\mathbf{I}$ is the identity matrix and $\mathbf{D}^{(2)} = (\delta_{ij}^2)$ is the matrix of the square dissimilarities for the training patterns. If $\mathbf{B}$ is positive semi-definite, the object coordinates in the low dimensional Euclidean space R^k can be found through

a singular value decomposition (Golub, 1996):

$$\mathbf{X}_k = \mathbf{V}_k \, \mathbf{\Lambda}_k^{1/2}, \tag{2}$$

where $\mathbf{V}_k \in R^{nxk}$ is an orthogonal matrix with columns the first k eigen vectors of $\mathbf{X}\,\mathbf{X}^T$ and $\mathbf{\Lambda}_k = \mathrm{diag}(\lambda_1 \ldots \lambda_k) \in R^{kxk}$ is a diagonal matrix with λ_i the i-th eigenvalue. Several dissimilarities introduced in section 2 generate inner product matrices B non semi-definite positive. Fortunately, the negative values are small in our application and therefore can be neglected without losing relevant information about the data (Pekalska, 2001).

Once the training patterns have been embedded into a low dimensional Euclidean space, the test pattern can be added to this space via a linear projection (Pekalska, 2001). Next we comment briefly the derivation.

Let $\mathbf{X}_k \in R^{nxk}$ be the object configuration for the training patterns in R^k and $\mathbf{X}_n = [x_1,.., x_s]^T \in R^{sxk}$ the matrix of the object coordinates sought for the test patterns. Let $\mathbf{D}_n^{(2)} \in R^{sxn}$ be the matrix of the square dissimilarities between the s test patterns and the n training patterns that have been already projected. The matrix $\mathbf{B}_n \in R^{sxn}$ of inner products among the test and training patterns can be found as:

$$\mathbf{B}_n = -½ \, (\mathbf{D}_n^{(2)} \, \mathbf{J} - \mathbf{U} \, \mathbf{D}^{(2)} \, \mathbf{J}) \tag{3},$$

where $\mathbf{J} \in R^{nxn}$ is the centering matrix and $\mathbf{U} = 1/n \, \mathbf{1}^T \, \mathbf{1} \in R^{sxn}$. The derivation of equation is detailed in (Pekalska, 2001). Since the matrix of inner products verifies

$$\mathbf{B}_n = \mathbf{X}_n \, \mathbf{X}_k^T, \tag{4}$$

then, $\mathbf{X}_n$ can be found as the least mean-square error solution to (4), that is:

$$\mathbf{X}_n = \mathbf{B}_n \, \mathbf{X}_k \, (\mathbf{X}_k^T \, \mathbf{X}_k)^{-1} \tag{5}$$

Given that $\mathbf{X}_k^T \, \mathbf{X}_k = \mathbf{\Lambda}_k$ and considering that $\mathbf{X}_k = \mathbf{V}_k \, \mathbf{\Lambda}_k^{1/2}$ the coordinates for the test points can be obtained as:

$$\mathbf{X}_n = \mathbf{B}_n \, \mathbf{V}_k \, \mathbf{\Lambda}_k^{-1/2}, \tag{6}$$

which can be easily evaluated through simple linear algebraic operations.

Figure 1. Aggregation of classifiers using a voting strategy. Bold patterns are missclassified by a single hyperplane but not by the combination.
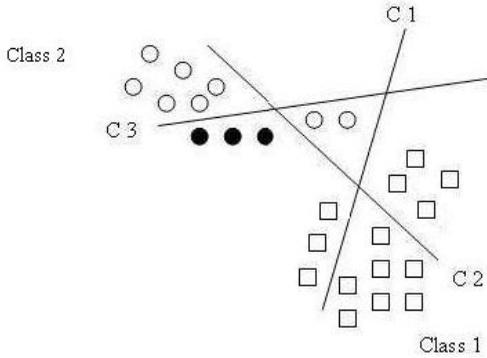
The combination strategy proposed here is based on the evidence that different dissimilarities reflect different features of the dataset (see section 2). Therefore, classifiers based on different measures will missclassify a different set of patterns.

Figure 1 shows for instance that bold patterns are assigned to the wrong class by only one classifier but using a voting strategy the patterns will be assigned to the right class.

Hence, our combination algorithm proceeds as follows: First, the dissimilarities introduced in section 2 are computed. Each dissimilarity is embedded into an Euclidean space, training and test pattern coordinates are obtained using equations (2) and (6) respectively. To increase the diversity of classifiers, once the dissimilarities are embedded a bootstrap sample of the patterns is drawn. Next, we train a SVM for each dissimilarity and bootstrap sample. Thus, it is expected that misclassification errors will change from one classifier to another. So the combination of classifiers by a voting strategy will help to reduce the misclassification errors.

A related technique to combine classifiers is the Bagging (Breiman, 1996 & Bauer, 1999). This method generates a diversity of classifiers that are trained using several bootstrap samples. Next, the classifiers are aggregated using a voting strategy. Nevertheless there are three important differences between bagging and the method proposed in this section.

First, our method generates the diversity of classifiers by considering different dissimilarities and thus will induce a stronger diversity among classifiers. A second advantage of our method is that it is able to work directly with a dissimilarity matrix. Finally, the combination of several dissimilarities avoids the problem of choosing a particular dissimilarity for the application we are dealing with. This is a difficult and time consuming task.

Notice that the algorithm proposed earlier can be easily applied to other classifiers such as the k-nearest neighbor algorithm that are based on distances.

EXPERIMENTAL RESULTS

In this section, the ensemble of classifiers proposed is applied to the identification of spam messages.

The spam collection considered is available from the UCI Machine learning database (Hastie, 2001). The corpus is made up of 4601 emails from which 39.4 % are spam and 60.6 % legitimate messages. The number of features considered to codify the emails is 57 and they are described in (Hastie, 2001).

Figure 2. Eigenvalues for the multidimensional scaling algorithm with the cosine dissimilarity
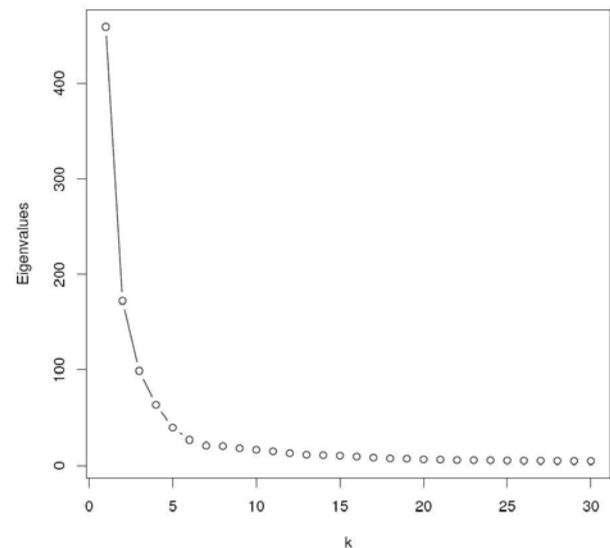
The dissimilarities have been computed without normalizing the variables because this may increase the correlation among them. Once the dissimilarities have been embedded in a Euclidean space, the variables are normalized to unit variance and zero mean. This preprocessing improves the SVM accuracy and the speed of convergence.

Regarding the ensemble of classifiers, an important issue is the dimensionality in which the dissimilarity matrix is embedded. To this aim, a metric Multidimensional Scaling algorithm is first run. The number of eigenvectors considered is determined by the curve induced by the eigenvalues. For the dataset considered, figure 2 shows that the first twenty eigenvalues preserve the main structure of the dataset.

The combination strategy proposed in this paper has been also applied to the k-nearest neighbor classifier. An important parameter in this algorithm is the number of neighbors which has been estimated using 20 % of the patterns as a validation set.

The classifiers have been evaluated from two different points of view: on the one hand we have computed the misclassification errors. But in our application, false positives errors are very expensive and should be avoided. Therefore false positive errors are also computed.

Table 1. Experimental results for the ensemble of SVM classifiers. Classifiers based solely on a single dissimilarity and Bagging have been taken as reference

	Linear Kernel		Polynomial Kernel	
Method	Error	False positive	Error	False positive
Euclidean	8.1%	4.0%	15%	11%
Cosine	19.1%	15.3%	30.4%	8%
Correlation	18.7%	9.8%	31%	7.8%
Manhattan	12.6%	6.3%	19.2%	7.1%
Kendall-τ	6.5%	3.1%	11.1%	5.4%
Spearman	6.6%	3.1%	11.1%	5.4%
Bagging Euclidean	7.3%	3.0%	14.3%	4%
Combination	**6.1%**	**3%**	**11.1%**	**1.8%**

Parameters: Linear kernel: C=0.1, m=20; Polynomial kernel:Degree=2, C=5, m=20

Table 2. Experimental results for the ensemble of k-NN classifiers. Classifiers based solely on a single dissimilarity and Bagging have been taken as reference

Method	Error	False positive
Euclidean	22.5%	9.3%
Cosine	23.3%	14.0%
Correlation	23.2%	14.0%
Manhattan	23.2%	12.2%
Kendall-τ	21.7%	6%
Bagging	19.1%	11.6%
Combination	**11.5%**	**5.5%**

Parameters: k=2

E

Finally the errors have been evaluated considering a subset of 20 % of the patterns drawn randomly without replacement from the original dataset.

Table 1 shows the experimental results for the ensemble of classifiers using the SVM. The method proposed has been compared with bagging introduced in section 3 and with classifiers based on a single dissimilarity.

From the analysis of table 1 the following conclusions can be drawn:

- The combination strategy improves significantly the Euclidean distance which is usually considered by most SVM algorithms.
- The combination strategy with polynomial kernel reduces significantly the false positive errors of the best single classifier. The improvement is smaller for the linear kernel. This can be explained because the non-linear kernel allow us to build classifiers with larger variance and therefore the combination strategy can achieve a larger improvement of the false positive errors. We also report that for the combination strategy as the C parameter increases the false positive errors converge to 0 although the false negative errors increase.
- The combination strategy proposed outperforms a widely used aggregation method such as Bagging. The improvement is particularly important for the polynomial kernel.

Table 2 shows the experimental results for the ensemble of k-NNs classifiers. As in the previous case, the combination strategy proposed improves particularly the false positive errors of classifiers based on a single distance. We also report that Bagging is not able to reduce the false positive errors of the Euclidean distance. Besides, our combination strategy improves significantly the Bagging algorithm. Finally, we observe that the misclassification errors are larger for k-NN than for the SVM. This can be explained because the SVM has a higher generalization ability.

CONCLUSIONS AND FUTURE RESEARCH TRENDS

In this paper, we have proposed an ensemble of classifiers based on a diversity of dissimilarities. Our approach aims to reduce particularly the false positive errors of classifiers based solely on a single distance. Besides, the algorithm is able to work directly from a dissimilarity matrix. The algorithm has been applied to the identification of spam messages.

The experimental results suggest that the method proposed help to improve both, misclassification errors and false positive errors. We also report that our algorithm outperforms classifiers based on a single dissimilarity and other combination strategies such as bagging.

As future research trends, we will try to apply other combination strategies that assign different weight to each classifier.

REFERENCES

Aggarwal, C. C., Re-designing distance functions and distance-based applications for high dimensional applications. *Proc. of the ACM International Conference on Management of Data and Symposium on Principles of Database Systems (SIGMOD-PODS)*, vol. 1, March 2001, pp. 13-18.

Androutsopoulos, I., Koutsias, J., Chandrinosand, K. V., Spyropoulos, C. D. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal E-mail Messages. *23rd Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval*, 160-167, Athens, Greece, 2000.

Bauer, E., Kohavi, R., An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, *Machine Learning*, vol.36, pp. 105-139, 1999.

Breiman, L. Bagging predictors, *Machine Learning*, vol. 24, pp. 123-140, 1996.

Carreras, X., Márquez, I. Boosting Trees for Anti-Spam Email Filtering. *RANLP-01, Forth International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, BG, 2001.

Cox, T., Cox, M. *Multidimensional Scaling*, 2nd ed. New York: Chapman & Hall/CRC Press, 2001.

Drucker, H., Wu, D. Vapnik, V. N. Support Vector Machines for Spam Categorization. *IEEE Transactions on Neural Networks*, 10(5), 1048-1054, September, 1999.

Fawcett, T. ``In vivo'' spam filtering: A challenge problem for KDD. *ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, 5(2), 140-148, December, 2003.

Golub, G.H., Loan, C.F.V. *Matrix Computations*, 3rd ed. Baltimore, Maryland, USA: Johns Hopkins University press, 1996.

Hastie, T., Tibshirani, R., Friedman, J. H. The Elements of Statistical Learning. Springer Verlag, Berlin, 2001. UCI Machine Learning Database. Available from: www.ics.uci.edu/~mlearn/MLRepository.html.

Hershkop, S., Salvatore, J. S. Combining Email Models for False Positive Reduction. *ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, 98-107, August, Chicago, Illinois, 2005.

Hinneburg, C. C. A. A., Keim, D. A. What is the nearest neighbor in high dimensional spaces? *Proc. of the International Conference on Database Theory (ICDT)*. Cairo, Egypt: Morgan Kaufmann, September 2000, pp. 506-515.

Kittler, J., Hatef, M., Duin, R., Matas, J. On combining classifiers, *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 228-239, March 1998.

Kolcz, A., Alspector, J. SVM-based filtering of E-mail Spam with content-specific misclassification costs. *Workshop on Text Mining (TextDM'2001)*, 1-14, San Jose, California, 2001.

Martín-Merino, M., Muñoz, A. A new Sammon algorithm for sparse data visualization. *International Conference on Pattern Recognition (ICPR)*, vol.1. Cambridge (UK): IEEE Press, August 2004, pp. 477-481.

Martín-Merino, M., Muñoz, A. Self organizing map and Sammon mapping for asymmetric proximities. *Neurocomputing*, vol. 63, pp. 171-192, 2005.

Pekalska, E., Paclick, P., Duin, R. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, vol. 2, pp. 175-211, 2001.

Provost, F., Fawcett., T. Robust Classification for Imprecise Environments. *Machine Learning*, 42, 203-231, 2001.

Valentini, G., Dietterich, T. Bias-variance analysis of support vector machines for the development of svm-based ensemble methods. *Journal of Machine Learning Research*, vol. 5, pp. 725-775, 2004.

Vapnik, V. *Statistical Learning Theory*. New York: John Wiley & Sons, 1998.

KEY TERMS

Bootstrap: Resampling technique based on several random samples drawn with replacement.

Dissimilarity: It is a measure of proximity that does not obey the triangle inequality.

Kernel: Non-linear transformation to a high dimensional feature space.

K-NN: K-Nearest Neighbor algorithm for classication purposes.

MDS: Multidimensional Scaling Algorithm applied for the visualization of high dimensional data.

SVD: Singular Value Decomposition. Linear algebra operation that is used by many optimization algorithms.

SVM: Support Vector Machines classifier.

UCE: Unsolicited Commercial Email, also known as Spam.

Evolutionary Algorithms in Discredibility Detection

Bohumil Sulc
Czech Technical University in Prague, Czech Republic

David Klimanek
Czech Technical University in Prague, Czech Republic

INTRODUCTION

Evolutionary algorithms are well known optimization techniques suitable for solving various kinds of problems (Ruano, 2005). The new application of evolutionary algorithms represents their use in the detection of biased control loop functions caused by controlled variable sensor discredibility (Klimanek, Sulc, 2005). *Sensor discredibility* occurs when a sensor transmitting values of the controlled variable provides inexact information, however the information is not absolutely faulty yet. Use of discredible sensors in control circuits may cause the real values of controlled variables to exceed the range of tolerated differences, whereas zero control error is being displayed. However, this is not the only negative consequence. Sometimes, sensor discredibility is accompanied with undesirable and hardly recognizable side effects. Most typical is an increase of harmful emission production in the case of combustion control (Sulc, Klimanek, 2005).

We have found that evolutionary algorithms are useful tools for solving the particular problem of finding a software-based way (co-called software redundancy) of sensor discredibility detection. Software redundancy is a more economic way than the usual hardware redundancy, which is otherwise necessary in control loop protection against this small, invisible control error occurrence.

Namely, the standard genetic algorithm and the simulated annealing algorithm have been successfully applied and tested to minimize the given cost function; by means of these algorithms newly developed method is able to detect controlled variable sensor discredibility. When applied to combustion processes, production of harmful emissions can be kept within accepted limits.

Used application of evolutionary algorithms inclusive terminology transfer reflecting this application area can serve as an explanatory case study helping readers in better understanding the way how the evolutionary algorithms operate.

BACKGROUND

The above-mentioned controlled variable sensor discredibility detection represents a specific part of the fault detection field in control engineering. According to some authors (Venkatasubramanian, Rengaswamy, 200 3, Korbic, 2004), fault detection methods are classified into three general categories: quantitative model-based methods, qualitative model-based methods, and process history based methods. In contrast to the mentioned approaches, where priori knowledge about the process is needed, for the controlled variable sensor discredibility detection it is useful to employ methods of evolutionary algorithms. The main advantage of such a solution is that necessary information about the changes in controlled variable sensor properties can be obtained with the help of evolutionary algorithms based on the standard process data – this is, in any case, acquired and recorded for the sake of process control.

In order to apply evolutionary algorithms to controlled variable sensor discredibility detection, a cost function was designed as a *residual function e* defined by the absolute value of difference between the sensor model output (y_m) and the real sensor output (y_{real}),

$$e = |y_{real} - y_m| \qquad (1)$$

The design of residual function *e* has been explained in detail (e.g. in Sulc, Klimanek, 2005).

In most sensor models it is assumed that the sensor output is proportional only to one input (Koushanfar, 2003), so that the sensor model equation is

$$y_m = k_m\, x_{est} + q_m, \qquad\qquad (2)$$

where parameter k_m represents the gain of the sensor model, parameter q_m expresses the shift factor, and x_{est} is the estimated sensor model input, which represents the physical (real) value of the control variable. The physical value of the control variable is not available for us because we expect that the sensor is not reliable and we want to detect this stage. However, we can estimate this value from the other process data that are acquired usually for the purposes of the information system. This estimation is usually based on steady-state data, so that it is important to detect the steady state of the process.

Basically, the underlying idea of applying the evolutionary algorithm is then based on finding a vector of the sensor model parameter for which the value of residual function e is minimal.

Advantages of the Evolutionary Algorithm Applied to Discredibility Detection

In principle, any optimization method could be used for the mentioned optimization task. The problem is that the sensor model input is an unknown, dynamically-changing variable. Therefore, the choice and the parameter selection must include certain element of a random selection from many alternatives, which is fulfilled in the case of evolutionary algorithms. The higher computational time requirements do not matter in the case of sensor discredibility detection, because the loss of credibility is the result of a gradual development.

Problem Statement

A particular task of evolutionary algorithms in the solved problem is e.g. a finding extreme of a given cost function. We have utilized the evolutionary algorithms to minimize the given cost function (in fault detection terminology a residual function). Based on this minimization, it is possible to detect that the control variable sensor is providing biased data.

THE STANDARD GENETIC ALGORITHM AND THE SIMULATED ANNEALING ALGORITHM IN DESCREDIBILITY DETECTION

Both methods have been tested and proved to be legitimate for use. Unlike general genetic presentations of the methods, we will present the methods in a transformed way, based on the use of terms from the field of fault detection. From the engineering view point this should facilitate understanding of both procedures (Klimanek, Sulc, 2005). In our text, the terms introduced in the theory of evolutionary algorithms are indicated by the abbreviation "ET".

The Standard Genetic Algorithm

In controlled variable sensor discredibility detection that uses genetic algorithm methods, the following steps are required (procedure by Fleming & Purshouse, 1995) (Figure 1):

1. Initialization – during initialization, the evolutionary time is set to zero and an initial set of vectors containing the sensor model parameters (called population in ET) is randomly generated within an expected range of reasonable values for each of the parameters. For each of the parameter vectors of the sensor model (in ET, individuals of the population), the value of the residual function

Figure 1 A flow chart of the standard genetic algorithm applied for discredibility detection
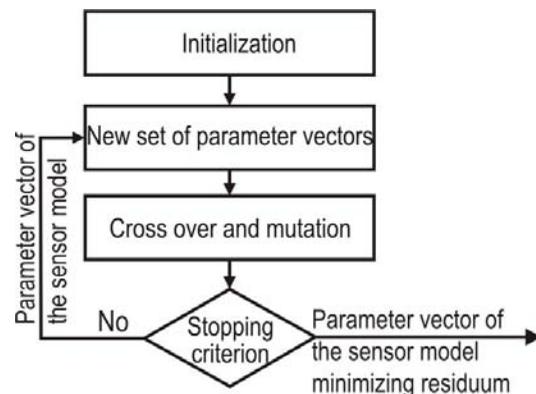
(1) is evaluated. Also, the average value of the residual function values is computed.

2. New set of parameter vectors – after starting the iteration process, a new set of parameter vectors is generated (in ET, new population) when the selection operator is employed. Selecting a set of the new parameter vectors, the following algorithm is used: the parameter vector of the sensor model that provides residual values lower than the average value is replicated into the next subset for generating new parameter candidates in more copies than in the original set, and the individuals with below-average residual values are rejected (Witczak, Obuchowicz, Korbicz, 2002).

3. Crossover and mutation (ET) – next comes the crossover operation over the randomly selected pairs of the parameter vector of the sensor model of the topical set. In the presenting application of the standard genetic algorithm, the selected sensor model parameters are coded into binary strings and the standard one point crossover operator is used. The mutation operator mimics random mutations (Fleming, Purshouse, 1995). The newly created parameters are coded into binary strings and one bit of each string is switched with random probability. The value of the residual function (1) for the current run is evaluated. Also, the average value of the residual function values is computed.

4. Stopping-criterion decision – if the stopping criterion is not met, a return to step 2 repeats the process. The stopping criterion is met, e.g. when the size of the difference between the average of the residual values from the current run and the average of the residual values from the previous run is lower then the given size (Fleming, Purshouse, 2002).

Simulated Annealing Algorithm

Controlled variable sensor discredibility detection via simulated annealing can be described by the following steps (Figure 2):

5. Initialization – an initial control parameter is set (in ET, initial annealing temperature). The control parameter is used to evaluate the Boltzmann criterion (King, 1999), which affects the acceptance of the current parameter vector of the sensor model

in step 3. A vector of random values of the sensor model parameters is selected and the value of the residual function is computed.

6. New set of parameter vectors – the iteration index is increased and, using a stochastic strategy, a new vector of the sensor model parameters is randomly generated (in ET it is spoken about generating new individuals) and the corresponding value of the residual variable is obtained (in ET, value of the cost function).

7. Boltzmann criterion computation – the difference between the residual value obtained in step 2 and the residual value from the previous iteration is evaluated. If the difference is negative, then the new parameter vector is accepted automatically. Otherwise, the algorithm may accept the new parameter vector based on the Boltzmann criterion. The control parameter is weighted with a coefficient λ (in ET, gradual temperature reduction). If the control parameter is less than or equal to the given final control parameter, then the stop criterion is met and the current vector of the sensor model parameters is accepted. Otherwise, returning to step 3 repeats the process of optimizing search.

Comparison of Usability of the Algorithms for Discredibility Detection

The comparison of both evolutionary algorithms applied to controlled variable sensor discredibility detection

Figure 2. A flow chart of the simulated annealing algorithm applied for dis-credibility detection
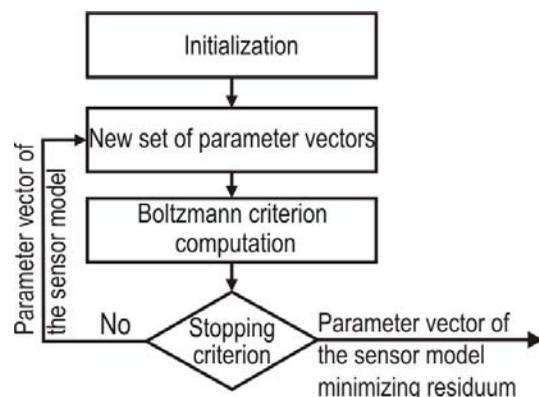
is shown by Figure 5. This comparison represents a part of results form paper Sulc, Klimanek. (2006). It is evident, that the simulated annealing algorithm needs more evaluation time for one evaluation period – a period for simulated annealing required 80 iterations, while genetic algorithm needed 40 iterations. This difference is because genetic algorithm works with a group of potential solutions, while simulated annealing compares only two potential solutions and accepts better one.

No difference was found between the two evolutionary algorithms used here; their good convergence depends mainly on the algorithm settings. Although evolutionary algorithms are generally much more time consuming than other optimizing procedures, this consideration does not matter in control variable sensor discredibility detection. This is because control variable

sensor discredibility has no conclusive impacts on the control results and the time needed for the detection does not affect the control process.

Testing Model-Based Sensor Discredibility Detection Method

The model-based control variable sensor discredibility detection method using evolutionary algorithms was tested to find whether the method is able to detect the control variable sensor properties changes via presented evolutionary algorithms. The simulation experiments are more described in (Klimanek, Sulc 2006a, Klimanek, Sulc 2006b). Results from the simulated experiments were summarized and they can be graphically demonstrated in the next paragraph.

Figure 3. Detection of gradual changes of the level sensor gain via genetic algorithm and the simulated annealing algorithm

E
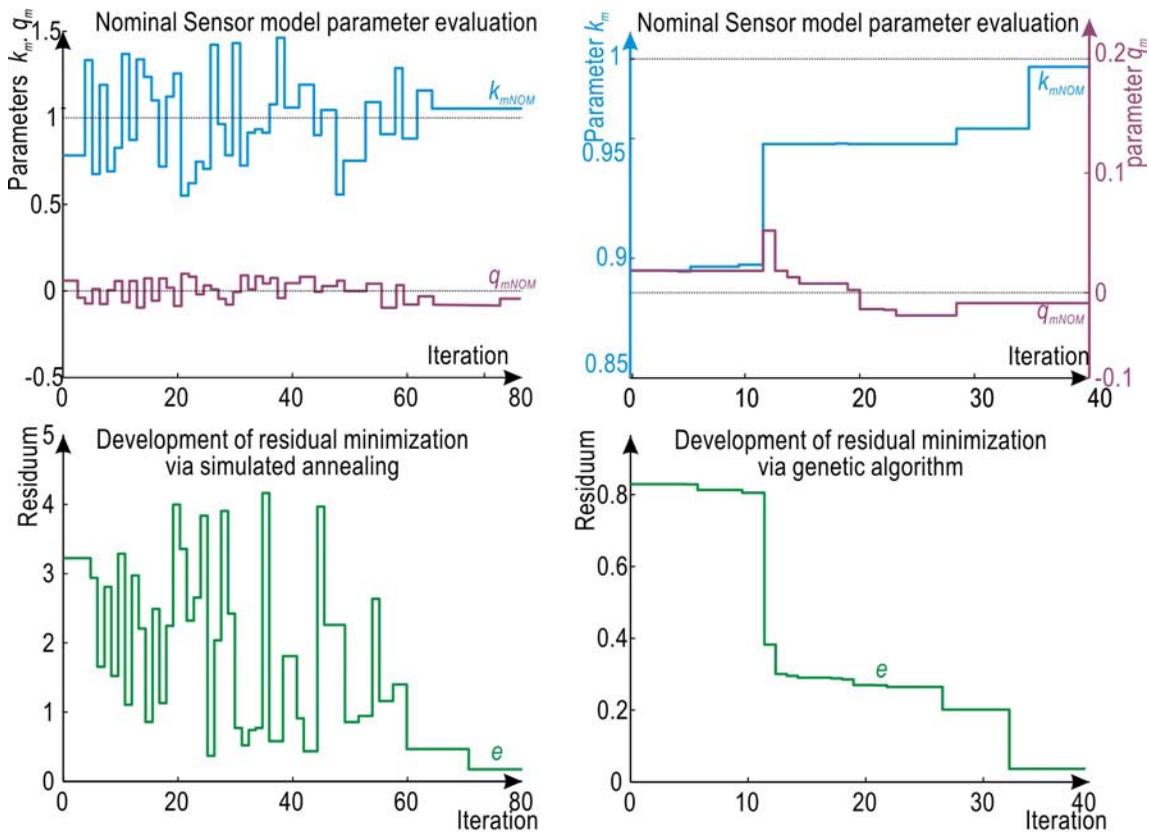
Results and Findings from the Tests

Figure 4 depicts a simulation run during which the sensor gain has been gradually decreased from a starting (correct) value. It can be seen that after the sensor properties has been changed, the measured value of the controlled variable (in this case the water level) is different from the correct value.

It is apparent that the algorithm used for sensor model parameter detection (in this case the genetic algorithm) is able to find the sensor model gain k_m, because the sensor model parameter development corresponds to the simulated real sensor parameter changing.

Figure 4. Detection of gradual changes of the level sensor gain via genetic algorithm
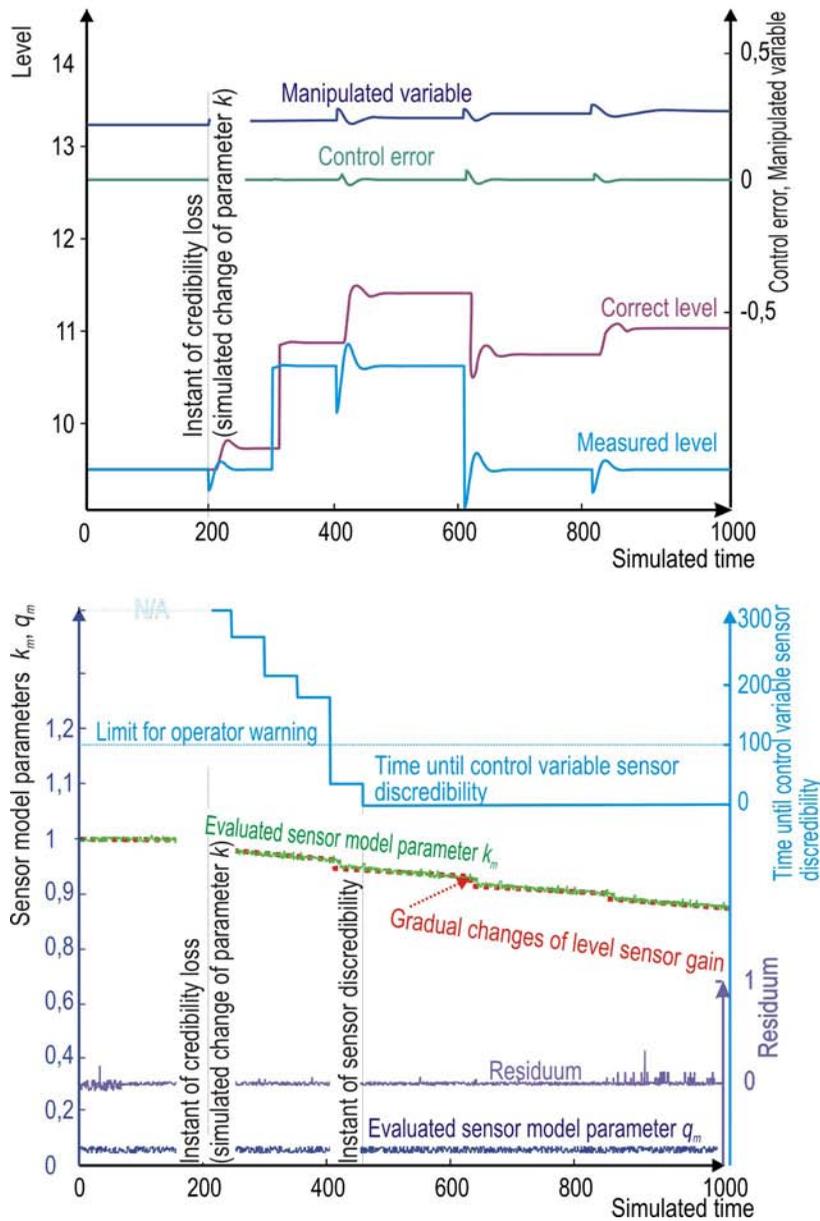
The sensor level discredibility detection results obtained using the simulated annealing algorithm, were similar. Figure 5 shows results obtained when the model-based method using the simulated annealing algorithm was tested. A step change of sensor gain was simulated and it is obvious that the algorithm was able to capture the change.

By this method the operator is informed about the estimated time remaining before sensor discredibility occurs. If the time is critical, the operator also receives timely a warning about the situation.

Figure 5. Detection of the step change of the level sensor gain via simulated annealing method

E
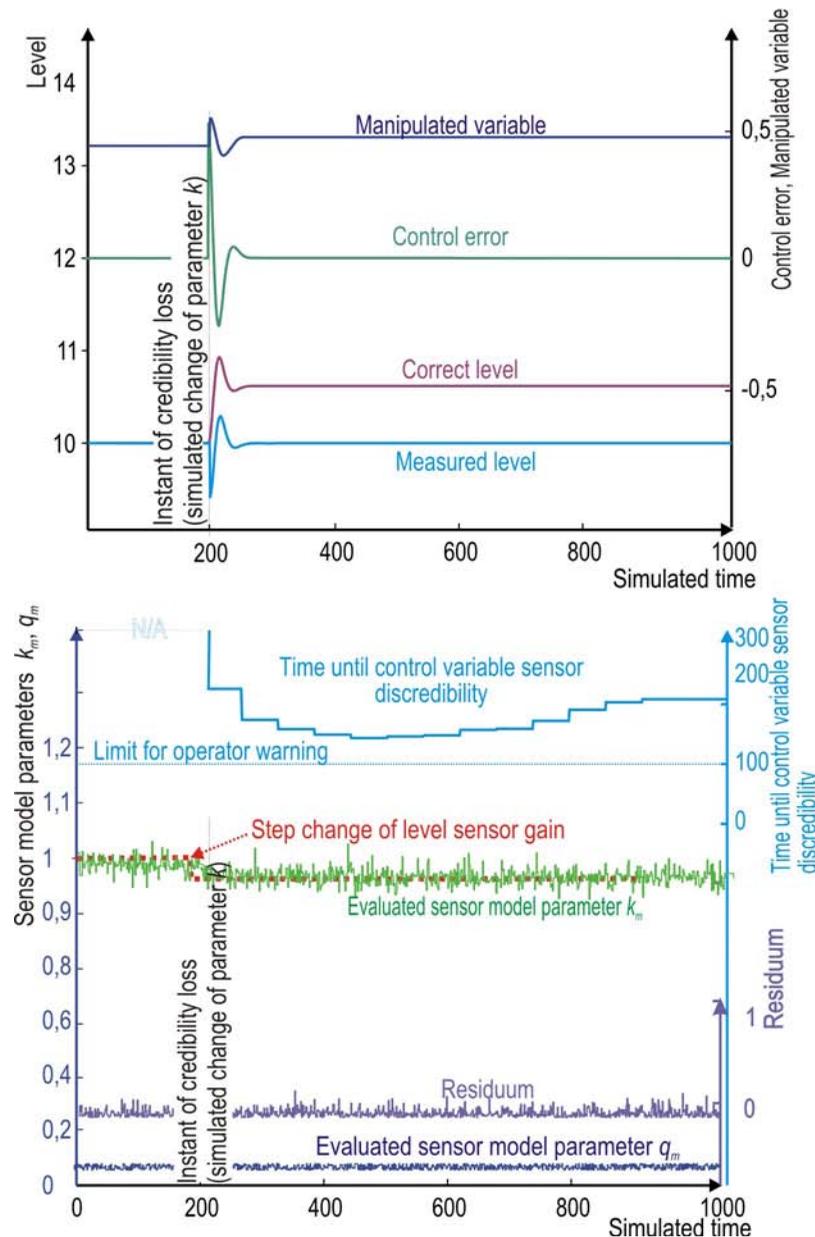
FUTURE TRENDS

It can be expected in future that tools for the controlled variable discredibility detection will be a standard accessory of any standard PID controller. On the present from economic reasons, the cases, when the described method using evolutionary algorithms is mostly justified to be implemented, are linked to revelation of undesirable side effects (increased production of harmful emissions CO, NO_x, or, in the case of bioenergetic processes, increase of unwanted production of CO_2). Except use for a standard detection of the changes in the controlled variable sensor, the discredibility detection provides possibilities to warn operator against occurrence of the control loop inaccuracy, or even to forecast it.

CONCLUSION

Implementation feasibility of the evolutionary algorithms in a model-based controlled variable sensor discredibility has been demonstrated here. In two variations of the standard evolutionary algorithms - the genetic algorithm and the simulated annealing algorithm, designed procedure of sensor discredibility detection was presented.

In both cases, the time needed for the evaluation was several minutes. In the case of the application for discredibility detection, this time demand does not matter because such small malfunctions do not lead to fatal errors in control loop operation and discredibility is usually a long developing process.

Evolutionary algorithms have become a useful tool in discovering hidden inaccuracy in the control loops. Discredibility detection saves costs on redundant controlled variable sensors, which are required if the controlled variable sensor discredibility is detected via hardware redundancy on the assumption that the costs for additional sensors are not negligible, of course.

The importance of discredibility detection using evolutionary algorithms can be found, e.g. in biomass combustion processes (due to the penalties for over-stepped limits in harmful emissions), and also in the food-processing industry, where side effects may not be harmful, but rather unpleasant (i.e. bad odors).

REFERENCES

Chen Z., He, Y., Chu F., Huang J. (2003). *Evolutionary Strategy for Classification Problems and its Application in Fault Diagnosis*. Engineering Application of Artificial Intelligence, Vol. 16, No. 1, 31–38.

Fleming P., Purshouse C. (1995). The Matlab Genetic Algorithm Toolbox. *IEE Colloquium on Applied Control Technology Using Matlab.* England, Sheffield.

Fleming P., Purshouse C (2002). *Evolutionary Algorithms in Control Systems Engineering: A survey.* Control Engineering Practice, Vol. 10, No. 11, 1223–1241.

King R. (1999). *Computational Methods in Control Engineering.* Kluwer, Academic Publishers.

Klimanek D., Sulc B. (2006a). Improvement of Control Loop Function by Control Variable Sensor Discredibility Detection, *Transactions of the VSB–Technical University of Ostrava.* Vol. 2, 120–126.

Klimanek D., Sulc B. (2006b). Sensor Discredibility Detection by Means of Software Redundancy. *Proceedings of the 7th International Carpathian Control Conference*, 249–252.

Klimanek D., Sulc,B. (2005). Evolutionary Detection of Sensor Discredibility in Control Loops, *Proceedings of the 31st Annual Conference IEEE.* 136–141.

Korbic J. (2004). *Fault Diagnosis: Models, Artificial Intelligence, Applications.* Berlin, Springer.

Koushanfar R. (2003). On-line fault detection of sensor measurements. *Proceedings of IEEE Sensors 2003,* Vol. 2, No. 8, 974–979.

Ruano A.E. (Ed.) (2005): *Intelligent Control Systems Using Computational Intelligence Techniques.* London: The IEE Press.

Sulc B., Klimanek D. (2005). Sensor Discredibility Detection via Computational Intelligence, *WSEAS Transactions on Systems.* Vol. 4, No. 11, 1906–1915.

Sulc B., Klimanek D. (2006). Enhanced Function of Standard Controller by Control Variable Sensor Discredibility Detection, *Proceedings of the WSEAS International Conferences: ACS06, EDU06, REMOTE06, POWER06, ICOSSSE06*, 119–124.

Venkatasubramanian V., Rengaswamy R. (2003). A Review of Process Fault Detection and Diagnosis. Quantitative Model-based Methods. *Computers & Chemical Engineering.* Vol. 27, No 3, 293–311.

Witczak M. (2003). *Identification and Fault Detection of Non-Linear Dynamic Systems*. Poland, University of Zielona Gora Press.

Witczak M., Obuchowicz A., Korbicz J. (2002). *Genetic Programming Based Approaches to Identification and Fault Diagnosis of Non-linear Dynamic Systems*. International Journal of Control, Vol. 75, No. 13, 1012–1031.

KEY TERMS

The following terms and definitions introduce a fault detection engineering interpretation of the terms usual in the evolutionary algorithm vocabulary. This should facilitate orientation in the presented engineering problem.

Chromosome: A particular sensor model parameter vector (a term for individuals used in evolutionary terminology).

Cost Function: A criterion evaluating level of the congruence between the sensor model output and the real sensor output. In the fault detection terminology, the cost function corresponds to the term residuum (or residual function).

Evolutionary Time: The number assigned to steps in the sequence of iteration performed during a search for sensor model parameters based on evolutionary development.

Individual: A vector of the sensor model parameters in a set of possible values (see population).

Initial Annealing Temperature: An initial algorithm parameter. Annealing temperature is used as a measure of evolutionary progress during the simulated annealing algorithm run.

Population: A set of the vectors of the sensor model parameters with which the sensor model has a chance to approach the minimum of the residual function.

Population Size: The number of the sensor model parameter vectors taken into the consideration in population.

Sensor Discredibility: A stage of the controlled variable sensor at which the sensor is not completely out of function yet, but its properties have gradually changed to the extend that the data provided by the sensor are so biased that the tolerated inaccuracy of the controlled variable is over-ranged and usually linked with possible side effects.

Evolutionary Approaches for ANNs Design

Antonia Azzini
University of Milan, Italy

Andrea G. B. Tettamanzi
University of Milan, Italy

INTRODUCTION

Artificial neural networks (ANNs) are computational models, loosely inspired by biological neural networks, consisting of interconnected groups of artificial neurons which process information using a connectionist approach.

ANNs are widely applied to problems like pattern recognition, classification, and time series analysis. The success of an ANN application usually requires a high number of experiments. Moreover, several parameters of an ANN can affect the accuracy of solutions. A particular type of evolving system, namely neuro-genetic systems, have become a very important research topic in ANN design. They make up the so-called Evolutionary Artificial Neural Networks (EANNs), i.e., biologically-inspired computational models that use evolutionary algorithms (EAs) in conjunction with ANNs.

Evolutionary algorithms and state-of-the-art design of EANN were introduced first in the milestone survey by Xin Yao (1999), and, more recently, by Abraham (2004), by Cantu-Paz and Kamath (2005), and then by Castellani (2006).

The aim of this article is to present the main evolutionary techniques used to optimize the ANN design, providing a description of the topics related to neural network design and corresponding issues, and then, some of the most recent developments of EANNs found in the literature. Finally a brief summary is given, with a few concluding remarks.

ARTIFICIAL NEURAL NETWORK DESIGN

In ANN design, the successful application of an ANN usually demands much experimentation. There are many parameters to set. Some of them involve ANN type, others the number of layers and nodes defining the architecture and the connection weights. Also the training data are an important factor, and a great deal of attention must be paid to the test data to make sure that the network will generalize correctly on data which has not been trained on.

Feature selection, structure design, and weight training can be regarded as three search problems in the discrete space of subsets of data attributes, the discrete space of the possible ANN configurations, and the continuous space of the ANN parameters, respectively.

Architecture design is crucial in the successful application of ANNs because it has a significant impact on their information-processing capabilities. Indeed, given a learning task, an ANN with only a few connections and linear nodes may not be able to perform the task at all, while an ANN with a large number of connections and nonlinear nodes may overfit noise in the training data and lack generalization. The main problem is that there is no systematic way to design an optimal architecture for a given task automatically.

Several methods have been proposed to overcome these shortcomings. This chapter focuses on one of them, namely EANNs. One distinct feature of EANNs is their adaptability to a dynamic environment. EANNs can be regarded as a general framework for adaptive systems, i.e., systems that can change their architectures and learning rules appropriately without human intervention.

In order to improve the performance of EAs, different selection schemes and genetic operators have been proposed in the literature. This kind of evolutionary learning for ANNs has also been introduced to reduce and, if possible, to avoid the problems of traditional gradient descent techniques, such as Backpropagation (BP), that lie in the trapping in local minima. EAs are known to be little sensitive to initial training conditions, due to their being global optimization methods, while a gradient descent algorithm can only find a local

optimum in a neighbourhood of the initial solution. EANNs provide a solution to these problems and an alternative for controlling network complexity.

ANN design can be regarded as an optimization problem. Tettamanzi and Tomassini (2001) presented a discussion about evolutionary systems and their interaction with neural and fuzzy systems, and Cantu-Paz and Kamath (2005) also described an empirical comparison of EAs and ANNs for classification problems.

EVOLUTIONARY ARTIFICIAL NEURAL NETWORKS

There are several approaches to evolve ANNs, that usually fall into two broad categories: *problem-independent* and *problem-dependent* representation of EAs. The former are based on a general representation, independent of the type and structure of the ANN sought for, and require the definition of an encoding scheme suitable for Genetic Algorithms (GAs). They can include mapping between ANNs and binary representation, taking care of decoders or repair algorithms, but this task is not usually easy.

The latter are EAs where chromosome representation is a specific data structure that naturally maps to an ANN, to which appropriate genetic operators apply.

EAs are used to perform various tasks, such as connection weight training, architecture design, learning rule adaptation, input feature selection, connection weight initialization, rule extraction from ANNs, etc. Three of them are considered as the most popular at the following levels:

- *Connection weights* concentrates just on weights optimization, assuming that the architecture of the network is given. The evolution of weights introduces an adaptive and global approach to training, especially in the reinforcement learning and recurrent network learning paradigm, where gradient-based training algorithms often experience great difficulties.
- *Learning rules* can be regarded as a process of "learning how to learn" in ANNs where the adaptation of learning rules is achieved through evolution. It can also be regarded as an adaptive process of automatic discovery of novel learning rules.

- *Architecture* enables ANNs to adapt their topologies to different tasks without human intervention. It also provides an approach to automatic ANN design as both weights and structures can be evolved. In this case a further subdivision can be made by defining a "pure" architecture evolution and a simultaneous evolution of both architecture and weights.

Other approaches consider the evolution of transfer functions of an ANN and input feature selection, but they are usually applied in conjunction with one of the three methods above in order to obtain better results.

The use of evolutionary learning for ANNs design is no more than two decades old. However, substantial work has been made in these years, whose main outcomes are presented below.

Weight Optimization

Evolution of weights may be regarded as an alternative training algorithm. The primary motivation for using evolutionary techniques instead of traditional gradient-descent techniques such as BP, as reported by Rumelhart et al. (1986), lies in avoiding trapping in local minima and the requirement that the activation function be differentiable. For this reason, rather than adapting weights based on local improvement only, EAs evolve weights based on the fitness of the whole network.

Some approaches use GAs with real encodings for biases and weights, like in the work presented by Montana and Davis (1989); others used binary weights encoding at first, and then implemented a modified version with real encodings as Whitley *et al.* (1990). Mordaunt and Zalzala (2002) implemented a real number representation to evolve weights, analyzing evolution with mutation and a multi-point crossover, while Seiffert (2001) described an approach to completely substitute a traditional gradient descent algorithm by a GA in the training phase.

Often, during the application of GAs, some problems, e.g., premature convergence and stagnation of solution can occur as reported by Goldberg (1992). In order to solve this problem, an improved algorithm was proposed by Yang et al. (2002), where a genetic algorithm, based on evolutionary stable strategy, was implemented to keep the balance between population diversity and convergence speed during evolution.

E

Recently, a new GA was proposed by Pai (2004), where a genetic inheritance operator was implemented to determine the weights of EANN, without considering mutation operators, but only two-point crossover for reproduction, applying it to decimal chromosomes.

Learning Rule Optimization

In supervised learning algorithms, standard BP is the most popular method for training multilayer networks. The design of training algorithms, in particular the learning rules used to adjust connection weights, depends on the type of ANN architecture considered. Several standard learning rules have been proposed, but designing an optimal learning rule becomes very difficult when there is little prior knowledge about the network topology, producing a very complex relationship between evolution and learning. The evolutionary approach becomes important in modelling the creative process since newly evolved learning rules can deal with a complex and dynamic environment.

The first kind of optimization considers the adjustment of learning parameters and can be seen as the first attempt to evolve learning rules. They comprise BP parameters, like the learning rate and momentum, and genetic parameters, like mutation and crossover probabilities. Some works have been carried out by Merelo et al. (2002), that presented several solutions for the optimal learning parameters of multilayer competitive-learning neural networks.

Considering learning-rule optimization, one of the first studies was conducted by Chalmers (1990). He also noticed that discovering complex learning rules using GAs is not easy, due to the highly complex genetic coding used, making the search space large and hard to explore, while GAs used a simpler coding which allows known learning rules as a possibility, making the search very biased. In order to overcome these limitations, Chalmers suggested to apply GP, a particular kind of GA. Several studies have been carried out in this direction and some of them are described, along with a new approach presented by Poli and Radi (2002).

Architecture Optimization

The design of an optimal architecture can be formulated as a search problem in the architecture space, where each point represents an ANN topology. As pointed out by Yao (1999), given some performance (optimality) criteria, e.g., minimum error, learning speed, lower complexity, etc., about architectures, the performance level of all these forms a surface in the design space. Several approaches have been carried out in this direction. A neuro-evolutionary approach was presented by Miikkulainen and Stanley (2002), using augmenting topologies. It has been designed specifically to outperform the solutions that employ a principled method of crossover of different topologies, to protect structural innovation using speciation, and to incrementally grow from minimal structure.

Another work carried out by Wang et al. (2002), considered the definition of an optimal network that was based on the combination of constructing and pruning by GAs, while, more recently, Bevilacqua et al. (2006) presented a multi-objective GA approach to optimize the search for the optimal topology, based on *Schema Theory*.

One of the most important forms of deception in ANNs structure optimization arises from the many-to-one and from one-to-many mapping from genotypes in the representation space to phenotypes in the evaluation space. The existence of networks functionally equivalent and with different encodings makes evolution inefficient. This problem is termed as the *competing convention problem*. Other important issues involve representation and the definition of the EA. In the encoding phase, an important aspect is to decide how much information about architecture should be encoded into the genotype. Then, the performance of ANNs strongly depends on their topology, considering size and structure, and, consequently, its definition characterizes networks features like its learning process speed, learning precision, noise tolerance, and generalization capability.

Transfer Function Optimization

Transfer function perturbations can begin with a fixed function, as linear, sigmoidal or gaussian, and allow the GA to adapt to a useful combination according to the situation. Some work has been carried out by Yao and Liu (1996) in order to apply a transfer function adaptation over generations, and by Figueira and Poli (1999), with a GP algorithm evolving functions.

To improve solutions, often, this kind of evolution is carried out together with the other kinds of ANNs optimizations, here described.

Input Data Selection

One of the most important factors for training neural networks is the availability and the integrity of data. They should represent all possible states of the problem considered, and they should have enough patterns for building also the test and validation set.

The consistency of all data has to be guaranteed, and the training data must be representative of the problem, in order to avoid overfitting.

Input data selection can be regarded as a search problem in the discrete space of the subsets of data attributes. The solution requires the removal of unnecessary, conflicting, overlapping and redundant features in order to maximize the classifier accuracy, compactness, and learning capabilities.

Input data reduction has been approached by Reeves and Taylor (1998), who applied genetic algorithms to select training sets for a kind of ANNs, and by Castellani (2006), embedding the search for the optimal feature set into the training phase.

Joint Evolution of Architecture and Weights

The drawbacks related to individual architecture and weights evolutionary techniques can be overcome with approaches that consider their conjunction.

The advantage of combining these two basic elements of an ANN is that a completely functioning network can be evolved without any intervention by an expert.

Several methods that evolve both the network structure and the connection weights were proposed in the literature.

Castillo et al. (1999) presented a method to search for the optimal set of weights, the optimal topology and learning parameters using a GA for the network evolution and BP for network training, while Yao et al. (1997, 2003) implemented, respectively, an evolutionary system for evolving feedforward ANNs based on evolutionary programming, and, more recently, a novel constructive algorithm for training cooperative NN ensembles. Azzini and Tettamanzi (2006) presented a neuro-genetic approach for the joint optimization of network structures and weights, taking advantage of BP as a specialized decoder, and Pedrajas et al. (2003) proposed a cooperative co-evolutionary method for ANN design.

OPEN ISSUES

There are still several open issues in EANNs research.

Regarding connection weights, a critical aspect is that the structure has to be predetermined, giving some problems when such a topology is difficult to define in the first place.

Also in learning rule evolution, the design of training algorithms, in particular the learning rules, depends on the type of the network architecture. Therefore, the design of such rules can become very difficult when there is little prior knowledge about the network topology, giving a complex relationship between evolution and learning.

The architecture evolution has an important impact on the neural network evolution, and the evolution of pure architecture presents difficulties in evaluating fitness accurately.

The simultaneous evolution of architecture and weights is one of the most interesting evolutionary ANNs techniques, and nowadays it concerns useful solutions for ANN design. Different works are carried out in these directions and are still open issues. Some of them concern about the application of cooperative or competitive co-evolutionary approaches, some others regarding the design of NN ensembles.

CONCLUSION

This work present a survey of the state of the art of evolutionary systems investigated in these decades and presented in the literature. In particular, this work focuses on the application of evolutionary algorithms to neural network design optimization.

Several approaches for NN evolution are presented, together with some related works, and for each method the most important features are presented together with their main advantages and shortcomings.

REFERENCES

Abraham, A. (2004). *Meta learning evolutionary artificial neural networks*. Neurocomputing, 56, 1-38.

Azzini, A. & Tettamanzi, A.G.B. (2006). *A neural evolutionary approach to financial modeling*. In Maarten

Keijzer et al. editor, Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2006. ACM Press, New York, NY.

Bevilacqua, V. & Mastronardi, G. & Menolascina, F. & Pannarale, P. & Pedone, A. (2006). *A Novel Multi-Objective Algorithm Approach to Artificial Neural Network Topology Optimisation: The Breast Cancer Classification Problem.* Proceedings of the International Joint Conference on Neural Networks, IJCNN'06, 1958-1965, Vancourver, Canada, IEEE Press.

Cantu-Paz, E & Kamath, C. (2005). *An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems.* IEEE Transactions on Systems, Management and Cybernetic, part B, 35 (5), 915-927.

Castellani, M. (2006). *ANNE - A New Algorithm for Evolution of Artificial Neural Network Classifier Systems.* Proceedings of the IEEE Congress on Evolutionary Computation - CEC 2006, 3294-3301, Vancouver, Canada.

Castillo, P.A. & Rivas, V. & Merelo, J.J. & Gonzalez, J. & Prieto, A. & Romero, G. (1999). *G-PropIII: global optimization of multilayer perceptrons using an evolutionary algorithm.* Proceedings of the Genetic and Evolutionary Computation Conference, 1, 942.6

Chalmers, D.J. (1990). *The evolution of learning: An experiment in genetic connectionism.* Proceedings of the Connectionist Models Summer School, 81-90.

Figueira, J.C. & Poli, R. (1999). *Evolution of neural networks using weight mapping.* Proceedings of the Genetic and Evolutionary Computation Conference, 2, 1170-1177.

Goldberg, D.E. (1992). *Genetic Algorithms in Search Optimization & Machine Learning.* Addison-Wesley.

Merelo, J.J & Castillo, P.A. & Prieto, A. & Rojas, I. & Romero, G. (2002). *Statistical analysis of the parameters of a neuro-genetic algorithm.* IEEE Transactions on Neural Networks, 13(6), 1374 - 1394.

Miikkulainen, R. & Stanley, K. (2002). *Evolving neural networks through augmenting topologies.* Evolutionary Computation, 10(2), 99-127.

Montana, D. & Davis, L. (1989). *Training feedforward neural networks using genetic algorithms.* Proceedin-gs of the 11th International Conference on Artificial Intelligence, 762-767.

Mordaunt, P. & Zalzala, A.M.S. (2002). *Towards an evolutionary neural network for gait analysis.* Proceedings of the Congress on Evolutionary Computation, 2, 1238-1243.

Pai, G.A. (2004). *A fast converging evolutionary neural network for the prediction of uplift capacity of suction caissons.* Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, 654-659.

Pedrajas, N.G. & Martinez, C.H. & Prez, J.M. (2003). *Covnet: A cooperative coevolutionary model for evolving artificial neural networks.* IEEE Transactions on Neural Networks, 14(3), 575-596.

Poli, R. & Radi, A. (2002). *Discovering efficient learning rules for feedforward neural networks using genetic programming.* Technical Report, University of Essex.

Reeves, C.R. & Taylor, S.J. (1998). *Selection of training data for neural networks by a genetic algorithm.* Parallel Problem Solving from Nature, Lecture Notes in Computer Science, 1498, 633-642.

Rumelhart, D.E. & McClelland, J.L. & Williams, R.J. (1986). *Learning representations by back-propagating errors.* Nature, 323, 533-536.

Seiffert, U. (2001). *Multiple layer perceptron training using genetic algorithms.* Proceedings of the European Symposium on Artificial Neural Networks, ESANN'2001, 159-164.

Tettamanzi, A.G.B. & Tomassini, M. (2001). *Soft computing: integrating evolutionary, neural, and fuzzy systems.* Springer-Verlag.

Wang, W. & Lu, W. & Leung, A.Y.T. & Lo, S. & Xu, Z. & Wang, X. (2002). *Optimal feed-forward neural networks based on the combination of constructing and pruning by genetic algorithms.* Proceedings of the International Joint Conference on Neural Networks, 636-641.

Whitley, D. & Starkweather, T. & Bogart, C. (1990). *Genetic algorithms and neural networks: Optimizing connections and connectivity.* Parallel computing, 14, 347-361.

Yang, B. & Su, X.H. & Wang, Y.D. (2002). *Bp neural network optimization based on an improved genetic algorithm.* Proceedings of the IEEE First International Conference on Machine Learning and Cybernetics, 64-68.

Yao, X. & Liu, Y. (1996). *Evolving artificial neural networks through evolutionary programming.* Evolutionary Programming V: *Proceedings of the Conference on Evolutionary Programming,* MIT Press, 257-266.

Yao, X. & Liu, Y. (1997). *A new evolutionary system for evolving artificial neural networks.* IEEE Transactions on Neural Networks, 8(3), 694-713.

Yao, X. (1999). *Evolving artificial neural networks.* Proceedings of the IEEE, 87, 1423-1447.

Yao, X. & Murase, K. & Islam, M.M. (2003). *A constructive algorithm for training cooperative neural network ensembles.* IEEE Transactions on Neural Networks, 14(4), 820-834.

KEY TERMS

Adaptive System: System able to adapt its behavior according to changes in its environment or in parts of the system itself.

Artificial Neural Networks: Models inspired by the working of the brain, considered as a combination of neurons and synaptic connections, which are capable of transmitting data through multiple layers, giving a system able to solve different problems like pattern recognition and classification.

Backpropagation Algorithm: A supervised learning technique used for training ANNs. It is based on a set of recursive formulas for computing the gradient vector of the error function, that can be used in a first-order method like gradient descent.

Error Backpropagation: Essentially a search procedure that attempts to minimize a whole network error function such as the sum of the squared error of the network output over a set of training input/output pairs.

Evolutionary Algorithms: Algorithms based on models that consider 'artificial' or 'simulated' genetic evolution of individuals in a defined environment. They are a broad class of stochastic optimization algorithms, inspired by biology and in particular by those biological processes that allow populations of organisms to adapt to their surrounding environment: genetic inheritance and survival of the fittest.

Evolutionary Artificial Neural Networks: Special class of artificial neural networks in which evolution is another fundamental form of adaptation in addition to learning. They are represented by biologically inspired computational models that use evolutionary algorithms in conjunction with neural networks to solve problems.

Evolutionary Computation: In computer science it is a subfield of artificial intelligence (more particularly computational intelligence) involving combinatorial optimization problems. Evolutionary computation defines the quite young field of the study of computational systems based on the idea of natural evolution and adaptation.

Multi-Layer Perceptrons (MLPs): Class of neural networks that consists of a feed-forward fully connected network with an input layer of neurons, one or more hidden layers and an output layer. The output value is obtained through the sequence of activation functions defined in each hidden layer. Usually, in this kind of network, the supervised learning process is the backpropagation algorithm.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in.

Evolutionary Approaches to Variable Selection

Marcos Gestal
University of A Coruña, Spain

José Manuel Andrade
University of A Coruña, Spain

INTRODUCTION

The importance of juice beverages in daily food habits makes juice authentication an important issue, for example, to avoid fraudulent practices.

A successful classification model should address two important cornerstones of the quality control of juice-based beverages: to monitor the amount of juice and to monitor the amount (and nature) of other substances added to the beverages. Particularly, sugar addition is a common and simple adulteration, though difficult to characterize. Other adulteration methods, either alone or combined, include addition of water, pulp wash, cheaper juices, colorants, and other undeclared additives (intended to mimic the compositional profiles of pure juices) (Saavedra, García, & Barbas, 2000).

VARIABLE SELECTON BY MEANS OF EVOLUTIONARY TECHNIQUES

This chapter presents several approaches to address the variable selection problem. All of them are based on evolutionary techniques. They can be divided into two groups. First group of techniques are based on different codifications of a traditional Genetic Algorithm (GA) population and different specifications for the evaluation function. Second group shows a modification in the traditional Genetic Algorithm to improve the generalization capability by adding a new population and an approach based on the evolution of subspecies into the genetic population.

BACKGROUND

A range of analytical techniques have been used to deal with authentication problems. These include high performance liquid chromatography (Yuan & Chen,

1999), gas chromatography (Stöber, Martin & Peppard, 1998) and isotopic methods (Jamin, González, Remaud, Naulet & Martin, 1997). Unfortunately, they are expensive and slow.

Infrared Spectrometry (IR) (Rodriguez-Saona, Fry, McLaughlin, & Calvey, 2001) is a fast and convenient technique to perform screening studies in order to assess the quantity of pure juice in commercial beverages. The interest lies in developing, from the spectroscopy data, classification methods that might enable the determination of the amount of natural juice contained in a sample.

However, the information gathered from the IR analyses has some fuzzy characteristics (random noise, unclear chemical assignment, etc.), so analytical chemists tend to use techniques like Artificial Neural Networks (ANN) (Haykin, 1999) or develop ad-hoc classification models. Previous studies (Gestal, Gómez-Carracedo, Andrade, Dorado, Fernández, Prada, & Pazos, 2005) showed that ANN classified apple juice beverages according to the concentration of natural juice they contained and that ANN had advantages over classical statistical methods, such as robust models and easy application of the methodology on R&D laboratories. Disappointingly, the large number of variables derived from IR spectrometry makes ANNs time-consuming during training and, most important, makes it very difficult to establish relationships between these variables and the analytical knowledge.

Several approaches were used to reduce the number of variables to a small subset, which should retain the classification capabilities of the overall dataset. Hence, the ANN training process and the interpretation of the results would be highly improved.

Furthermore, previous variable selection would yield other advantages: cost reduction (if the classification model requires a reduced set of data, the time needed to obtain them will be shorter; increased efficiency (if the system processes less information, less

time for processing it will be required); understanding improvement (if two models resolve the same task, but one of them uses less information this would be more thoroughly interpreted. Therefore, the simpler the model, the easier the knowledge extraction and the easier the understanding, the easier the validation).

In addition, it was proved the analysis of IR data involved a highly multimodal problem, as many combinations of variables each (obtained using a different method) led to similar results when the samples were classified.

GENETIC ALGORITHMS

A GA (Holland, 1975)(Goldberg, 1989) is a recurrent and stochastic process that operates with a group of potential solutions to a problem, known as genetic population, based on one of the Darwin's principles: the survival of the best individuals (Darwin, 1859).

Briefly a GA works as follows. Initially, a population of solutions is generated randomly and the solutions evolve continuously after consecutive stages of crossovers and mutations. Every individual at the population has an associated value that quantifies associated its usefulness (adjustment or fitness), in accordance to its adequacy to solve the problem. This value has to be obtained for each potential solution and constitutes the quantitative information the evolutionary algorithm will use to guide the search. The process will continue until a predetermined stopping criterion is reached. This might be a particular threshold error for the solution or a certain number of generations (populations).

Therefore, different basic steps will be required to implement a GA: codification of the problem, which results in a population structure, initialisation of the first population, defining a fitness function to evaluate how good is each individual to solve the problem and, finally, a cyclic procedure of reproductions and replacements (Michalewicz, 1999; Goldberg, 2002).

DATA DESCRIPTION

In the present practical application, the spectral range measured by IR spectrometry (wavenumbers from 1250 cm-1 to 900 cm-1) provided 176 absorbances (which measured light absorption)(Gómez-Carracedo, Gestal, Dorado & Andrade, 2007).

The main goal of the application consisted on the prediction of the amount of pure juice on a sample using absorbance values returned for the IR measurements. But the amount of data obtained for a sample by IR spectrometry is huge, so the direct application of mathematical and/or computational methods (although possible) requires a lot of time. Accordingly, it is important to establish whether all raw data provided relevant information for sample differentiation. Hence, the problem was an appropriate case for the use of variable selection techniques.

Previous to variable selection construction of data sets for both model development and validation was required. Thus, samples with different amounts of pure apple juice were prepared at the laboratory. Besides, 23 apple juice-based beverages sold in Spain were analysed (the declared amount of juice printed out on

Table 1. Low and high concentrations dataset

Juice Concentration	2%	4%	6%	8%	10%	16%	20%	Total
Training	19	17	16	22	21	20	19	134
Validation	1	1	13	6	6	6	6	39
Commercial	0	0	0	0	1	1	0	2

Juice Concentration	20%	25%	50%	70%	100%	Total
Training	20	19	16	14	7	86
Validation	6	18	13	1	6	44
Commercial	0	2	0	0	19	21

their labels was used as input data). The samples were distributed in 2 ranges: samples containing less than 20% of pure juice and samples with more than 20% of pure apple juice (Table 1). IR spectra were obtained for all the samples.

This data was split in two datasets to extract the rules (ANN training) and to validate them. The commercial samples were used to further check the performance of the model. It is worth noting that whenever a predicted value does not match that given on the labels of the commercial products it might be owing to either a true wrong performance of the model (classification error) or an inaccurate labelling of the commercial beverage.

Classification Test Considering all the Original Variables

First test involved all variables given by IR spectroscopy. A dedicated ANN used all the absorbances of the training data to obtain a classification model. Later, the classification results obtained with this model will be used as a reference to compare the performance of the proposals over the same data.

Different parametric classification techniques (PLS, SIMCA, Potential Curves, etc.) were used too (Gestal, Gómez-Carracedo, Andrade, Dorado, Fernández, Prada, & Pazos, 2004), with very similar results. But the best results were achieved using ANN, which will be very useful to address the variable selection issue employing GA with fitness functions based on ANN. The accuracy of the reference method is shown in Table 2.

An exhaustive study of the results of the different classification models allowed us to conclude that the

set of samples with low (2-20%) percentages of juice was far more complex and difficult to classify than the samples with higher percentages (25-100%). Indeed, the number of errors was usually higher for the 2-20% range both in calibration and validation. Classification of the commercial samples agreed quite fine with the percentages of juice declared into the labels, but for a particular sample. When that sample was studied in detail it was observed that its spectrum was slightly different from the usual ones. This suggested that the juice contained an unusually high amount of added sugar(s).

VARIABLE SELECTION APPROACHES

Variable selection processes were performed to optimize the ANN classifications.

First, two simple approaches will be briefly described: Pruned and Fixed Search. Both approaches are based on a traditional GA and both use ANN to evaluate fitness. As the results will show, both techniques offer good solutions but present a common problem: an execution of each method provides only a solution (discarding any other possible optimal one). This was addressed with the two more advanced approaches described in the next sections.

Regardless of the variable selection approach, a GA will guide the search by evaluating the prediction capabilities of each ANN model developed employing different sets of IR variables. The problem that has to be solved is to find out a small set of IR variables that, when combined with an ANN model, classifies

Table 2. Classification with ANNs using all the variables

Low Concentrations	Training (134)	Validation (39)	Comercial (2)
	134 (100%)	**35 (89.74%)**	**2 (100%)**

ANN Configuration for low concentrations
Topology: 176 / 50 / 80 / 5   learning rate: 0.0005   stop criterion: mse=5 or epochs=500.000

High Concentrations	Training (86)	Validation (44)	Commercial (21)
	86 (100%)	**43 (97.72%)**	**21 (100%)**

ANN Configuration
Topology: 176 / 8 / 5 / 5   learning rate: 0.0005   stop criterion: mse=1 or epochs=500.000

apple juice-based beverages properly (according to the amount of pure apple juice they contain).

Any time a subset of IR variables is proposed, the associated absorbance values are used as input patterns to the ANN. So, ANN will consider as much input processing elements (PEs) as variables. The output layer has one PE per category (6 for the lower range and 5 for the high ones). After several previous trials considering several hidden layers (from 1 to 4), each with different PEs (from 1 to 50), a compromise was established between the final fitness level reached by the ANN and the time required to its training. This compromise was essential because although better results were obtained with more hidden layers the time required for training was much higher as well. Then, it was decided not to extensively train the net but to get a good approximation to its real performance and elucidate whether the input variables are really suitable to accurately classify the samples.

The goal is to determine which solutions, among those provided by the GA, represent good starting points to perform more exhaustive training. Therefore, it would be enough to extend the ANN learning up to the point where it starts to converge. For this particular problem, convergence started after 800 cycles; to warrant it, 1000 iterations were fixed.

Next, each of the most promising solutions are used as inputs to an external ANN intended to provide the final classification results.

Pruned Search

This approach starts by considering all variables where from groups of variables are gradually discarded. The GA will steadily reduce the amount of variables that characterise the objects, until an optimal subset that allows for an overall satisfactory classification is obtained. This is used to classify the samples, and the results are used to determine how relevant the discarded wavenumbers were for the classification. This process can be continued as long as the classification results are equal, or at least similar, to those obtained using the overall set of variables. Therefore, the GA determines how many and which wavenumbers will be selected for the classification.

In this approach, each individual in the genetic population is described by n genes, each representing one variable. With a binary encoding each gene might be either 0 or 1, indicating whether the gene is active or not and, therefore, if the variable should be considered for classification.

The evaluation function has to guide the pruning process on getting individuals with a low number of variables. To achieve this, the function should help those individuals that, besides classifying accurately, make use of fewer variables. In this particular case a factor proportional to the percentage of active genes was defined to multiply the MSE obtained by the ANN, so that individuals with less active genes – and with a similar classification performance – will have a higher fitness and consequently a higher probability of survival.

Table 3 shows the results of several runs of Pruned Search approach. It is worth noting that each solution was extracted from a different execution and that, within the same execution, only one solution provided valid classification rates. As it can be noted, classification results were very similar although the variables used to perform the classifications were different.

These ANN models were obtained are slightly worse than those obtained using 176 wavenumbers, although the generalization capabilities of the best ANNs model were quite satisfactory as there was only one error when the commercial beverages were classified.

Fixed Search

This approach uses a real codification in the chromosome of the genetic individuals. The genetic population consists of individuals with n genes, where n is the amount of wavenumbers that are considered sufficient for the classification according to some external criterion. Each gene represents one of the 176 wavenumbers considered in the IR spectra. The GA will find out the n-variables subsets yielding the best classification models. The number of variables is predefined in the genotype.

As the final number of variables has to be decided in advance, some external criterion is needed. In order to simplify comparison of the results, the final number of variables was defined by the minimum number of principal components which can describe our data set, they were two.

Since the amount of wavenumbers remains constant in the course of the selection process, this approach defines the fitness of each genetic individual as the mean square error reached by the ANN at the end of the training process.

Table 3. Classification with pruned search

Low Concentrations	Training (134)	Validation (39)	Comercial (2)
Run1: Selected Variables [42 77]	129 (96.27%)	23 (58.97%)	1 (50%)
Run2: Selected Variables [52 141]	115 (85.82%)	22 (56.41%)	0 (0%)
Run3: Selected Variables [102 129]	124 (92.54%)	25 (64.10%)	0 (0%)
ANN Configuration for low concentrations *Topology: 2/ 10 / 60 / 7*	*learning rate: 0.0001*	*stop criterion: mse=5 or epochs=500.000*	
High Concentrations	**86**	**43**	**21**
Run1: Selected Variables [42 77]	83 (96.51%)	34 (79.07%)	21 (100%)
Run2: Selected Variables [52 141]	82 (95,35%)	35 (81.40%)	20 (95.24%)
Run3: Selected Variables [102 129]	83 (96.51%)	33 (76.74%)	21 (100%)
ANN Configuration *Topology: 2 / 10 / 60 / 5*	*learning rate: 0.0001*	*stop criterion: mse=1 or epochs=500.000*	

Table 4 shows the results of several runs of Pruned Search approach. Again, note that each run provides only a solution.

A problem was that the genetic individuals contains only two genes. Hence, any crossover operator will use the unique available crossing point (between the genes) and only half of the information from each parent will be transmitted to its offspring. This converts the Fixed Search approach into a "random search" when only two genes constituted the chromosome.

Hybrid Two-Population Genetic Algorithm

The main disadvantage of non-multimodal approaches is that they discard local optimal solutions because a final or global solution is preferred. But there are situations where the final model has to be extracted after analysing different similar solutions of the same problem.

For example, after analysing the different solutions provided by one execution of the classification task with Hybrid Two-Population Genetic Algorithm (Rabuñal, Dorado, Gestal & Pedreira, 2005) it was obtained three valid (and similar) models (Table 5). Furthermore the results were clearly superior to those obtained with the previous alternatives. But, the most important, it was observed that the solutions concentrated along specific spectral areas (around the 88 wavenumber). It would not be possible with the previous approaches. This approach will be studied deeply on a dedicated chapter (Finding multiple solutions with GA in multimodal problems) and no more details will be presented here.

Table 4. Classification with fixed search

Low Concentrations	Training (134)	Validation (39)	Comercial (2)
Run1: Selected Variables [12 159]	125 (93.28%)	23 (58.97%)	0 (0%)
Run2: Selected Variables [23 67]	129 (96.27%)	26 (66.66%)	1 (50%)
Run3: Selected Variables [102 129]	129 (96.27%)	25 (64.10%)	0 (0%)
ANN Configuration for low concentrations *Topology: 2/ 10 / 60 / 7*	*learning rate: 0.0001*	*stop criterion: mse=5 or epochs=500.000*	
High Concentrations	**86**	**43**	**21**
Run1: Selected Variables [12 159]	82 (95.35%)	31 (72.09%)	19 (90.47%)
Run2: Selected Variables [23 67]	81 (94.18%)	31 (72.09%)	19 (90.47%)
Run3: Selected Variables [102 129]	81 (94.18%)	33 (76.74%)	21 (100%)
ANN Configuration *Topology: 2 / 10 / 60 / 5*	*learning rate: 0.0001*	*stop criterion: mse=1 or epochs=500.000*	

FUTURE TRENDS

A next natural stage should be to consider multimodal approaches. Evolutionary Computation provides useful tools like fitness sharing, crowding… which should be compared to the hybrid two populations approach. Research is needed to implement criteria to allow the GA to stop when a satisfactory low number of variables is found.

Another suitable option should be include more scientific information within the system to guide the search. For example, a final user would provide a description of the ideal solution in terms of efficiency, data acquisition cost, simplicity, etc. All these parameters may be used as targets in a multiobjective Genetic Algorithm intended to provide the best variable subset complying with all the requirements.

CONCLUSION

Several conclusions can be drawn for variable selection tasks:

First, satisfactory classification results can be obtained from reduced sets of variables extracted using quite different techniques, all based on the combination of GA and ANN.

Best results were obtained using a multimodal GA (the hybrid two population approach), as it was expected, based on its ability to maintain the genetic individuals homogeneously distributed over the search space. Such diversity not only induces the appearance of optimal solutions, but also avoids the search to stop on a local minimum. This option does not provide only a solution but a group of them, with similar fitness. This allows scientists to select a solution with a sound chemical background and extract additional information.

Table 5. Classification with hybrid two-population genetic algorithm

Low Concentrations	Training (134)	Validation (39)	Comercial (2)
Run1: Selected Variables [89 102]	127 (95.77%)	29 (74.36%)	0 (0%)
Run1: Selected Variables [87 102]	130 (97.01%)	28 (71.79%)	0 (0%)
Run1: Selected Variables [88 89]	120 (89.55%)	29 (74.36%)	0 (0%)
ANN Configuration for low concentrations *Topology: 2/ 10 / 60 / 7*	*learning rate: 0.001*	*stop criterion: mse=2 or epochs=500.000*	
High Concentrations	86	43	21
Run1: Selected Variables [89 102]	83 (96.51%)	35 (81.39%)	21 (100%)
Run1: Selected Variables [87 102]	83 (96.51%)	36 (83.72%)	21 (100%)
Run1: Selected Variables [88 89]	82 (95.35%)	39 (90.69%)	21 (100%)
ANN Configuration *Topology: 2 / 10 / 60 / 5*	*learning rate: 0.001*	*stop criterion: mse=2 or epochs=500.000*	

REFERENCES

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection.*

Gestal, M., Gómez-Carracedo, M.P., Andrade, J.M., Dorado, J., Fernández, E., Prada, D., Pazos, A. (2005). Selection of variables by Genetic Algorithms to Classify Apple Beverages by Artificial Neural Networks. *Applied Artificial Intelligence.* 181-198.

Gestal, M., Gómez-Carracedo, M.P., Andrade, J.M., Dorado, J., Fernández, E., Prada, D., Pazos, A. (2004). Classification of Apple Beverages using Artificial Neural Networks with Previous Variable selection. *Analytica Chimica Acta.* 225-234.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Kluwer Academic Publishers, Boston.

Goldberg, D. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms.* Addison-Wesley, Reading, MA.

Gómez-Carracedo, M.P., Gestal, M., Dorado, J., & Andrade, J.M. (2007). Linking chemical knowledge and Genetic Algorithms using two populations and focused multimodal search. Chemometrics and intelligent laboratory systems. (87) 173-184.

Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation, 2nd Edition.* Prentice Hall.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press, ANN Arbor

Jamin, E., González, J., Remaud, G., Naulet, N. & Martin, G. (1997). Detection of exogenous sugars or organic acids addition in pineapple juices and concentrates by 13C IRMS analysis. *Journal of Agricultural and Food Chemestry.* (45), 3961-3967.

Michalewicz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag.

Rabuñal, J.R., Dorado, J., Gestal, M., & Pedreira, N. (2005). Diversity and Multimodal Search with a Hybrid Two-Population GA: an Application to ANN Development. *Lecture Notes in Computer Science.* 382-390.

Rodriguez-Saona, L.E, Fry, F.S., McLaughlin, M.A., & Calvey, E.M. (2001). Rapid analysis of sugars in fruit juices by FT-NIR spectroscopy. *Carbohydrate Research*. (336) 63-74.

Saavedra, L., García, A., Barbas, C. (2000). Development and validation of a capillary electrophoresis method for direct measurement of isocitric, citric, tartaric and malic acids as adulteration markers in orange juice. *Journal of Chromatography*. (881)1-2, 395-401.

Stöber, P., Martin, G.G., & Peppard, T.L. (1998). Quantitation of the undeclared addition of industrially produced sugar syrups to fruit by juices capillary gas chromatography. *Deutsche Lebensmittel-Rundschau*. (94) 309-316.

Yuan, J.P. & Chen, F. (1999). Simultaneous separation and determination of sugars, ascorbic acid and furanic compounds by HPLC-dual detection. *Food Chemistry*. (64) 423–427.

KEY TERMS

Absorbance: Function (usually logarithmic) of the percentage of transmission of a wavelength of light through a liquid.

Artificial Neural Network: Interconnected group of artificial neurons that uses a mathematical or computational model for information processing. They are based on the function of biologic neurons. It involves a group of simple processing elements (the neurons) which can exhibit complex global behaviour, as result of the connections between the neurons.

Evolutionary Technique: Technique which provides solutions for a problem guided by biological principles such as the survival of the fittest. These techniques start from a randomly generated population which evolves by means of crossover and mutation operations to provide the final solution.

Knowledge Extraction: Explicitation of the internal knowledge of a system or set of data in a way that is easily interpretable by the user.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in. Combination of all the possible values for all the variables related with the problem.

Spectroscopy (Spectrometry): Production, measurement and analysis of electromagnetic spectra produced as a result of the interactions between electromagnetic radiation and matter, such as emission or absorption of energy.

Spectrum: Intensity of a electromagnetic radiation across a range of wavelengths. It represents the intensity of emitted or transmitted energy versus the energy of the received light.

Variable Selection: Selection of a subset of relevant variables (features) which can describe a set of data.

Evolutionary Computing Approach for Ad–Hoc Networks

Prayag Narula
University of Delhi, India

Sudip Misra
Yale University, USA

Sanjay Kumar Dhurandher
University of Delhi, India

INTRODUCTION

Wireless ad-hoc networks are infrastructureless networks in which heterogeneous capable nodes assemble together and start communicating without any backbone support. These networks can be made truly dynamic and the nodes in these networks can move about freely while connecting and disconnecting with other nodes in the network. This property of ad-hoc networks to self-organize and communicate without any extrinsic support gives them tremendous flexibility and makes them perfect for applications such as emergencies, crisis-management, military and healthcare.

For example, in case of emergencies such as earthquakes, often most of the existing wired network infrastructure gets destroyed. In addition, since most of the wireless networks such as GSM and IEEE 802.11 wireless LAN use wired infrastructure as their backbone, often they are also rendered useless. In such scenarios, ad-hoc networks can be deployed swiftly and used for coordinating relief and rescue operations. Ad-hoc networks can be used for communication between various stations in the battle-field, where setting up a wired or an infrastructure-based network is often considered impractical.

Though a lot of research has been done on ad-hoc networks, a lot of problems such as security, quality-of-service (QoS) and multicasting need to be addressed satisfactorily before ad-hoc networks can move out of the labs and provide a flexible and cheap networking solution.

Evolutionary computing algorithms are a class of bio-inspired computing algorithms. Bio-inspired computing refers to the collection of algorithms that use techniques learnt from natural biological phenomena and implement them to solve a mathematical problem (Olario & Zomaya, 2006). Natural phenomena such as evolution, genetics, and collective behavior of social organisms and functioning of a mammalian brain teach us a variety of techniques that can be effectively employed to solve problems in computer science which are inherently tough.

In this Chapter and the chapter entitled, "Swarm Intelligence Approach for Wireless Ad Hoc Networks" of this book, we present some of the currently available important implementations of bio-inspired computing in the field of ad-hoc networks. This chapter looks at the problem of optimal clustering in ad-hoc networks and its solution using Genetic Programming (GP) approach. The chapter entitled, "Swarm Intelligence Approaches for Wireless Ad Hoc Networks" of this book, continues the same spirit and explains the use of the principles underlying Ant Colony Optimization (ACO) for routing in ad-hoc networks.

BACKGROUND

The first infrastructureless network was implemented as packet radio (Toh, 2002). It was initiated by the Defense Advanced Research Projects Agency (DARPA) in 1970s. By this time the ALOHA project (McQuillan & Walden, 1977) at the University of Hawaii had demonstrated the feasibility of using broadcasting for sending / receiving the data packets in single-hop radio networks. ALOHA later led to the development of Packet Radio Network (PRNET), which was a multi-hop multiple-access network, under the sponsorship of Advanced Research Projects Agency (ARPA). PRNET had the design objectives similar to

the current day ad-hoc networks such as flow and error control over multi-hop communication route, deriving and maintaining network topology information and mechanism to handle router mobility and power and size requirements, among others. However, since the electronic devices were huge then, the packet radios were not easily movable, leading to limited mobility. In addition, the network coverage was slow and since Bellman-Ford's shortest path algorithm was used for routing, transient loops were present. Since then, a lot of research has been done on ad-hoc networks and a number of routing algorithms have been developed which provide far greater performances and are loop free. The rapid development of silicon technology has also led to ever shrinking devices with increasing computation power. Ad-hoc networks are deliberated for use in medical, relief-and-rescue, office environments, personal networking, and many other daily life applications.

Bio-inspired algorithms, to which the evolutionary computing approaches such as genetic algorithms belong, have been around for more than past 50 years. In fact, there have been evidences that suggest that the Artificial Neural Networks are rooted in the unpublished works of A. Turing (related to the popular *Turing Machine*) (Paun, 2004). Finite Automata Theory was developed about a decade after that based on the neural modeling. This ultimately led to the area that is currently known as Neural Computing. This can effectively be called as the initiation of bio-inspired computing. Since then, techniques such as GP, Swarm Intelligence and Ant Colony Optimization and DNA computing have been researched and developed as nature continues to inspire us and show us the way for solving the most complex problems known to man.

MAIN FOCUS OF THE CHAPTER

This chapter and the the chapter entitled, "Swarm Intelligence Approaches for Wireless Ad Hoc Networks" of this book, in combination, present an introduction to various bio-inspired algorithms and describe their implementation in the area of wireless ad-hoc networks. This chapter primarily presents the GP approach to ad-hoc networks. We first give a general introduction to GP and explain the concepts of genes and chromosomes. We also explain the stochastic nature of GP and the process of mutation and crossover to provide optimal

solution to the problem using the GP approach. We then present the Weighted Clustering Algorithm given by (Chatterjee, Das & Turgut, 2002) that are used for clustering of nodes in mobile ad-hoc networks, as an instantiation of this approach.

GP

GP is a popular bio-inspired computing method. The concepts in genetic algorithms are inspired by the phenomenon of life and evolution itself. Life is a problem whose solution includes retaining those who have strong enough characteristics to survive in the environment and discarding the others. This exquisite process can provide solutions to complex analytical problems awaiting the most "fitting" result.

The basics include the role of chromosomes and genes. Chromosomes carry the genes which contain the parameters/characteristics that need to be optimized. Hence, GP starts with declaration of data structures that form the digital chromosomes. These digital chromosomes contain genetic information where each gene represents a parameter to be optimized. The gene could be represented as a single bit, which could be '1' (ON) or '0' (OFF). So, a chromosome is a sequence of 1's and 0's and a parameter is either totally present or totally absent. Other abstractions could represent the presence of a parameter in relative levels. For instance, a gene could be represented using 5 bits where the magnitude of the binary number tells about the magnitude of the presence of a parameter in the range '0' (00000) to '31' (11111).

First, these digital chromosomes are created using stochastic means. Then their fitness is tested either by static calculation of fitness using some method or dynamically by modelling fights between the chromosomes. The chromosomes with a set level of fitness are retained and allowed to produce a new generation of chromosomes. This can be done either by genetic recombination that is new chromosomes are produced with combination of present chromosomes, or by mutation, that is new chromosomes are produced by randomly producing changes in present chromosomes. This process of testing for fitness and creating new generations is repeated until the fittest chromosomes are deemed as optimized enough for the task, which the genetic algorithm was created for. The process is described in Figure 1.

E

Figure 1. Genetic algorithm
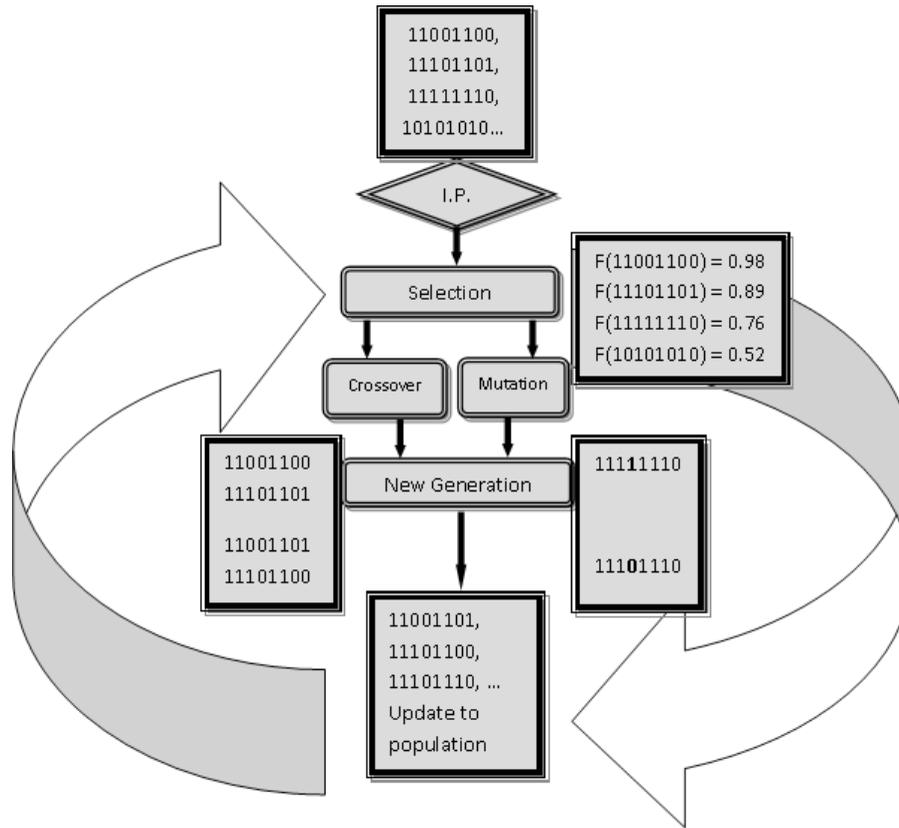
Genetic algorithms begin with a stochastic process and arrive at an optimized solution and are time consuming.. Hence, they are generally used for solving complex problems. As mentioned by Ashby (1962), self organization is one of these complex problems. Self organization is the problem where the components of a multi-component system achieve (or try to achieve) a common goal without any centralized or distributed control. The organization is generally done by changing the direct environment which can be adapted by the various system components and hence affect the behaviour of these components.

As has been mentioned, "Self-organization is especially important in ad-hoc networking because of the spontaneous interaction of multiple heterogeneous components over wireless radio connections without human interaction" (Murthy & S., 2004).

Dressler (2006) gives the following list of self organization capabilities:

• Self-healing: The system should be able to detect and repair the failures cause by overloading, component malfunctioning or system breakdown.
• Self-configuration: The system should be able to generate adequate configurations including connectivity, quality of service etc. as required in the existing scenario
• Self-management: The system should be able to maintain devices and in turn the network depending on the set configuration parameters.
• Self-optimization: The system should be to make an optimal choice of methods depending on the system behaviour .

- Adaptation: The system should dynamically adapt to the changing environment conditions, for example, change in node positions, change in number of nodes in the network etc.

Genetic algorithms have been used extensively in robotics, electronic circuit design, natural language processing, game theory, multi-model search, computer network topology design among many other applications.

In the context of wireless ad-hoc networks, genetic algorithms have been used in solving shortest path routing problem (Ahn & Ramakrishna, 2002), QoS path discovery (Fu, Li & Zhang, 2005), developing broadcasting strategy (Alba, Dorronsoro, Luna, Nebro & Bouvry, 2005), QoS routing (Barolli, Koyama & Shiratori, 2003), among others. In the interest of brevity, we present below only one representative application of the use of genetic algorithms to solve problems in ad-hoc networks.

Weighted Clustering Using Genetic Algorithm

Nodes in ad-hoc networks are sometimes grouped into various clusters with each cluster having a cluster-head. Clustering aims at introducing a kind of organisation in ad-hoc networks. This leads to better scalability of networks resulting in better utilisation of resources in larger networks. A cluster-head is responsible for the formation of clusters and maintenance of clusters in ad-hoc networks. Several clustering mechanisms have been proposed for ad-hoc networks, like Lowest-ID (Ephremides, Wieselthier & Baker, 1987), Highest Connectivity (Gerla & Tsai, 1995) Distributed Mobility-Adaptive Clustering (DMAC) (Basagni, 1999), Distributed Dynamic Clustering Algorithm (McDonald & Znati, 1999) and Weight-Based Adaptive Clustering Algorithm (WBACA) (Dhurandher & Singh, 2007).

Weighted Clustering Algorithm (WCA) (Chatterjee, Das and Turgut, 2002) is a popular clustering algorithm which selects a cluster-head on the basis of node mobility, battery-power, connectivity, distance from neighbour and degree of connectivity. Weights are assigned to each parameter and a combined weighted metric W_v is calculated as shown by Equation (1) (Chatterjee, Das and Turgut, 2002). Within certain constraints, nodes with minimum W_v are selected as the cluster-head.

$$W_v = w_1\Delta_v + w_2D_v + w_3M_v + w_4P_v \qquad (1)$$

In Equation (1),

Δ_v signifies the difference between optimal and actual connectivity of a node.

D_v signifies the sum of distances with all the neighbouring nodes.

M_v is the running average of the speed of the node.

P_v signifies the time that the node has acted as a cluster-head.

w_1, w_2, w_3 and w_4 are relative weights given to different parameters.

It should be noted that (Chatterjee, Das and Turgut, 2002):

$$w_1 + w_2 + w_3 + w_4 = 1 \qquad (2)$$

A genetic algorithm-based approach was presented by the designers of WCA (Chatterjee, Das, & Turgut, 2002). The sub-optimal solutions are mapped to chromosomes and given as input to produce best solutions using genetic techniques. This leads to better performance and more evenly balanced load sharing. The basic building blocks of the algorithm are given below:

Initial Population: A candidate solution which acts as a chromosome can be represented as shown in Figure 6. This initial population set is generated randomly by arranging the nodes in strings and then traversing this string. A node which is not a cluster-head and is not a neighbour of a cluster-head (and hence, part of an existing cluster) is chosen as a cluster head if it has less than δ (a pre-defined constant) neighbours. δ is chosen such as to prevent a cluster-head with more than optimal neighbour causing over-loading at cluster-heads.

Objective Fitness Function: Fitness value of a chromosome can be calculated as the sum of W_v of the contained genes. All nodes present at a gene [1] are analysed. If a node is not a cluster-head or a member of a cluster-head and has a node degree less than MAX_DEGREE, it is added to the cluster-head list and its W_v value is added to the total sum. For remaining nodes in the network, if the node is not a cluster-head or a member of other cluster-head, its W_v value is added to the sum and

E

the node is added to the cluster-head list. Lesser the sum of the W_v values of the genes, the higher is the fitness value of the chromosome.

Crossover: The crossover rate is 80 %. The authors used a technique called X_Over1 (Chatterjee, Das, & Turgut, 2002) as the crossover technique for the genetic implementation.

Mutation: Mutation introduces randomness into the solution space. If mutation rate is low, there is a chance of the solution converging to a non-optimal

Figure 2. Candidate solution as chromosome and its single gene (Based On: Turgut, Das, Elmasri, & Turgut, 2002, Fig. 3)
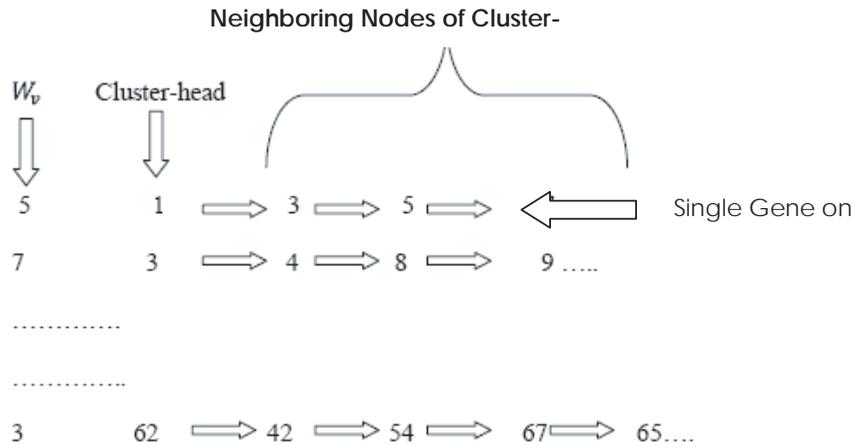
Figure 3. Genetic algorithm for weighted clustering in ad-hoc networks[1]

```
Algo_genetic_cluster{
        Generate initial population randomly, with population size = number of
nodes
do{
                while(new_pool_size < old_pool_size){
                                select chromosomes using Roulette wheel
method;
                                apply crossover using X_Over1;
                                apply mutation by swapping genes;
                                find fitness value of all the chromosomes;
                                replace through appending;
```

solution space. Hence, mutation is a very important step in genetic computations. In the stated algorithm, the inventors used swapping for the process of mutation. Two genes of a chromosome were selected randomly and swapped. Mutation rate used is 10%.

Selection Criteria: As mentioned earlier, chromosomes with lower values of W_v are considered fitter. Roulette wheel method is used for selection in accordance with the fitness values of these chromosomes.

Elitism: If the new generation produced has fitness value better than the best fitness value of the previous generation, than the best solution is replaced by the new generation. Since the best solutions of a generation are replaced, this step helps avoid the local maxima of the solution space and move towards the global maxima.

Replacement: This method states that, during replacement, the best solution of a generation is appended to the solution set of the next generation. This step helps in preserving the best solution during genetic operations.

Using these building functions, the genetic algorithm for the weighted clustering algorithm was given. This algorithm is given in Figure 3.

CONCLUSION

This article presented an overview of an evolutionary computing approach using GP and their application to wireless ad-hoc networks, both of which are currently "hot" topics amongst the computer science and networking research community. The intention of writing this article was to show how one could "marry" together concepts from GP with ad-hoc networks to arrive at interesting results. In particular we reviewed the WCA algorithm (Chatterjee, Das and Turgut, 2002) which uses GP for clustering of nodes in ad-hoc networks.

REFERENCES

Ahn, C. W., & Ramakrishna, R. S. (2002). A Genetic Algorithm for Shortest Path Routing and the sizing of population. *IEEE Transactions on Evolutionary Computation, Vol. 6, No. 6,.* IEEE Computer Society.

Alba, E., Dorronsoro, B., Luna, F., Nebro, A. J., & Bouvry, P. (2005). A Cellular Multi-Objective Genetic Algorithm for Optimal Broadcasting Strategy in Metropolitan MANETs. *19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05),* IEEE Computer Society Press.

Ashby, W. R. (1962). Principles of the Self-organizing system. In H. V. Foerster, & G. W. Zopf, *Principles of Self-Organization* (pp. 255–278). Eds. Pergamon Press.

Barolli, L., Koyama, A., & Shiratori, N. (2003). A QoS Routing Method for Ad-Hoc Networks Based on Genetic Algorithm. *14th International Workshop on Database and Expert Systems Applications(DEXA)* (pp. 175-179). Prague: IEEE Computer Society.

Basagni, S. (1999), Distributed Clustering for ad hoc networks, in *Proceedings of International Symposium on Parallel Architectures, Algorithms, and Networks (IS-PAN)*, Perth/Fremantle, Australia.

Chatterjee, M., Das, S., & Turgut, D. (2002). WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. *Cluster Computing* , 193-204

Dhurandher S. K., and Singh G.V. (2007), Stable Clustering with Efficient Routing in Wireless Ad Hoc Networks, *in Proceedings of IEEE International Conference on COMmunication System softWAre and MiddlewaRE (IEEE COMSWARE 2007)*.

Dressler, F. (2006). *Self-Organization in Ad Hoc Networks: Overview and Classification.* University of Erlangen, Dept. of Computer Science.

Ephremides, A., Wieselthier, J. E. and Baker, D. J. (1987), A Design Concept of Reliable Mobile Radio Networks with Frequency Hopping Signaling, *in Proceedings of the IEEE*, Vol. 75, No.1, pp. 56-73.

Ephremides, A., Wieselthier, J. E. and Baker, D. J. (1987), A Design Concept of Reliable Mobile Radio Networks with Frequency Hopping Signaling, *in Proceedings of the IEEE*, Vol. 75, No.1, pp. 56-73.

Fu, P., Li, J., & Zhang, D. (2005). Heuristic and Distributed QoS Route Discovery for Mobile Ad hoc Networks. *The Fifth International Conference on Computer and Information Technology (CIT'05).* IEEE Computer Society.

E

McDonald, A. B., and Znati, T. F. (1999), A mobility-based framework for adaptive clustering in wireless ad hoc networks, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1466-1487.

McQuillan, J. M., & Walden, D. C. (1977). The ARPA Network Design Decisions. *Computer Networks , I*, 243-289.

Murthy, C. S., & S., M. B. (2004). *Ad Hoc Wireless Networks.* Upper Saddle River, NJ: Prentice Hall.

Olario, S., & Zomaya, A. Y. (2006). *Handbook of Bioinspired Algorithms and Applications.* Chapman and Hall / CRC Press.

Paun, G. (2004). Bio-Inspired Computing Paradigms (Natural Computing). *Pre-Proc. Unconventional Programming Paradigms* (pp. 155-160). Berlin: Springer.

Toh, C. K. (2002). *Ad Hoc Mobile Wireless Systems.* Prentice Hall PTR.

Turgut, D., Das, S., Elmasri, R., & Turgut, B. (2002). Optimizing parameters of a mobile ad hoc network protocol with a genetic algorithm. *Global Telecommunications Conference* (pp. 62 - 66). IEEE Computer Society.

KEY TERMS

Bio-Inspired Algorithms: Group of algorithms modelled on the observed natural phenomena which are employed to solve mathematical problems.

Chromosome: A proposed solution to a problem which is represented as a string of bits and can mutate and cross-over to create a new solution.

Clustering: Grouping the nodes of an ad hoc network such that each group is a self-organized entity having a cluster-head which is responsible for formation and management of its cluster.

Cross-Over: Genetic operation which produces a new chromosome by combining the genes of two or more parent chromosome.

Fitness Function: A function which maps the subjective property of a fitness of a solution to an objective value which can be used to arrange different solutions in the order of their suitability as final or intermediate solution.

Genes: Genes are building blocks of chromosomes and represent the parameters that need to be optimized.

Genetic Algorithms: The algorithms that are modelled on the natural process of evolution. These algorithms employ methods such as crossover, mutation and natural selection and provide the best possible solutions after analyzing a group of sub-optimal solutions which are provided as inputs.

Initial Population: Set of sub-optimal solutions which are provided as inputs to a genetic algorithm and from which an optimal solution evolves.

Mobile Ad-Hoc Network: A multi-hop network formed by a group of mobile nodes which co-operate among each other to achieve communication, without requiring any supporting infrastructure.

Mutation: Genetic operation which randomly alters a chromosome to produce a new chromosome adding new solution to the solution-set.

ENDNOTE

[1]   Based on Chatterjee, Das and Turgut, 2002

Evolutionary Grammatical Inference

Ernesto Rodrigues
Federal University of Technology, Brazil

Heitor Silvério Lopes
Federal University of Technology, Brazil

INTRODUCTION

Grammatical Inference (also known as grammar induction) is the problem of learning a grammar for a language from a set of examples. In a broad sense, some data is presented to the learner that should return a grammar capable of explaining to some extent the input data. The grammar inferred from data can then be used to classify unseen data or provide some suitable model for it.

The classical formalization of **Grammatical Inference** (GI) is known as Language Identification in the Limit (Gold, 1967). Here, there are a finite set S_+ of strings known to belong to the language L (the positive examples) and another finite set S_- of strings not belonging to L (the negative examples). The language L is said to be identifiable in the limit if there exists a procedure to find a grammar G such that $S_+ \subseteq L(G)$, $S_- \not\subset L(G)$ and, in the limit, for sufficiently large S_+ and S_-, $L = L(G)$. The disjoint sets S_+ and S_- are given to provide clues for the inference of the production rules P of the unknown grammar G used to generate the language L.

Grammatical inference include such diverse fields as speech and natural language processing, gene analysis, pattern recognition, image processing, sequence prediction, information retrieval, cryptography, and many more. An excellent source for a state-of-the art overview of the subject is provided in (de la Higuera, 2005).

Traditionally, most work in GI has been focused on the inference of regular grammars trying to induce finite-state automata, which can be efficiently learned. For context free languages some recent approaches have shown limited success (Starckie, Costie & Zaanen, 2004), because the search space of possible grammars is infinite. Basically, the parenthesis and palindrome languages are common test cases for the effectiveness of grammatical inference methods. Both languages are context-free. The parenthesis language is deterministic but the palindrome language is nondeterministic (de la Higuera, 2005).

The use of evolutionary methods for context-free grammatical inference are not new, but only a few attempts have been successful.

Wyard (1991) used Genetic Algorithm (GA) to infer grammars for the language of correctly balanced and nested parentheses with success, but fails on the language of sentences containing the same number of a's and b's ($a^n b^n$ language). In another attempt (Wyard, 1994), he obtained positive results on the inference of two classes of context-free grammars: the class of n-symbol palindromes with $2 \leq n \leq 4$ and a class of small natural language grammars.

Sen and Janakiraman (1992) applied a GA using a pushdown automata to the inference and successfully learned the $a^n b^n$ language and the parentheses balancing problem. But their approach does not scale well.

Huijsen (1994) applied GA to infer context-free grammars for the parentheses balancing problem, the language of equal numbers of a's and b's and the even-length 2-symbol palindromes. Huijsen uses a "markerbased" encoding scheme with has the main advantage of allowing variable length chromosomes. The inference of regular grammars was successful but the inference of context-free grammars failed.

Those results obtained in earlier attempts using GA to context-free grammatical inference were limited. The first attempt to use Genetic Programming (GP) for grammatical inference used a pushdown automata (Dunay, 1994) and successfully learned the parenthesis language, but failed for the $a^n b^n$ language.

Korkmaz and Ucoluk (2001) also presented a GP approach using a prototype theory, which provides a way to recognize similarity between the grammars in the population. With this representation, it is possible to recognize the so-called building blocks but the results are preliminary.

Javed and his colleagues (2004) proposed a Genetic Programming (GP) approach with grammar-specific heuristic operators with non-random construction of the initial grammar population. Their approach succeeded in inducing small context-free grammars.

More recently, Rodrigues and Lopes (2006) proposed a hybrid GP approach that uses a confusion matrix to compute the fitness. They also proposed a local search mechanism that uses information obtained from the sentence parsing to generate a set of useful productions. The system was used for the parenthesis and palindromes languages with success.

BACKGROUND

A formal language is usually defined as follows. Given a finite alphabet Σ of symbols, we define the set of all strings (including the empty string ε) over Σ as Σ^*. Thus, we want to learn a language $L \subset \Sigma^*$. The alphabet Σ could be a set of characters or a set of words. The most common way to define a language is based on grammars which gives rules for combining symbols and to produce the all sentences of a language.

A grammar is defined by a quadruple $G = (N, \Sigma, P, S)$, where N is an alphabet of nonterminal symbols, Σ is an alphabet of terminal symbols such that $N \cap \Sigma = \phi$, P is a finite set of production rules of the form $\alpha \rightarrow \beta$ for $\alpha, \beta \in (N \cup \Sigma)^*$ where * represents the set of symbols that can be formed by taking any number of them, possibly with repetitions. S is a special nonterminal symbol called the start symbol.

The language $L(G)$ produced from grammar G is the set of all strings consisting only of terminal symbols that can be derived from the start symbol S by the application of production rules. The process of deriving strings by applying productions requires the definition of a new relation symbol $\Rightarrow$. Let $\alpha X \beta$ be a string of terminals and nonterminals, where X is a nonterminal. That is, α and β are strings in $(N \cup \Sigma)^*$, and $X \in N$. If $X \rightarrow \varphi$ is a production of G, we can say $\alpha X \beta \Rightarrow \alpha \varphi \beta$. It is important to say that one derivation step can replace any nonterminal anywhere in the string. We may extend the $\Rightarrow$ relationship to represent one or many derivation steps. We use a * to denote more steps. Therefore, we formally define the language $L(G)$ produced from grammar G as $L(G) = \{ w \mid w \in \Sigma^*, \ S \Rightarrow^* w \}$.

More details about formal languages and grammars can be found in textbooks such as Hopcroft et al (2001).

The Chomsky Hierarchy

Grammars are classified according to the form of the production rules used. They are commonly grouped into a hierarchy of four classes, known as the **Chomsky hierarchy** (Chomsky, 1957).

- *Recursively enumerable languages:* a grammar is unrestricted, and its productions may replace any number of grammar symbols by any other number of grammar symbols. The productions are of the form $\alpha \rightarrow \beta$ with $\alpha, \beta \in (N \cup \Sigma)^*$.
- *Context-sensitive languages:* they have grammars with productions that replace a single nonterminal by a string of symbols, whenever the nonterminal occurs in a specific *context*, i.e., has certain left and right neighbors. These productions are of the form $\alpha A \gamma \rightarrow \alpha \beta \gamma$, with $A \in N$ and $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$. A is replaced by β if it occurs between α and γ.
- *Context-free languages:* in this type, grammars have productions that replace a single nonterminal by a string of symbols, regardless of this nonterminal's context. The productions are of the form $A \rightarrow \alpha$ for $A \in N$ and $\alpha \in (N \cup \Sigma)^*$; thus A has no context.
- *Regular languages:* they have grammars in which a production may only replace a single nonterminal by another nonterminal and a terminal. The productions are of the form $A \rightarrow B\alpha$ or $A \rightarrow \alpha B$ for A, B $\in N$ and $\alpha \in \Sigma^*$.

It is sometimes useful to write a grammar in a particular form. The most commonly used in grammatical inference is the Chomsky Normal Form. A CFG G is in Chomsky Normal Form (CNF) if all production rules are of the form $A \rightarrow BC$ or $A \rightarrow \alpha$ for $A, B, C \in N$ and $\alpha \in \Sigma$.

The Cocke-Younger-Kasami Algorithm

To determine whether a string can be generated by a given context-free grammar in CNF, the Cocke-Younger-Kasami (CYK) algorithm can be used. This

algorithm is efficient and it has complexity $O(n^3)$ where n is the sentence length .

In the CYK algorithm, first a triangular table that tells whether the string w is in $L(G)$ is constructed. The horizontal line corresponds to the positions of the string $w = a_1 a_2 .. a_n$. The table entry V_{rs} is the set of variables A $\in P$ such that A $\Rightarrow^* a_r a_{r+1} ... a_s$. We are interested in whether the start symbol S is in the set V_{1n} because that is the same as saying $S \Rightarrow^* a_1 a_2 ... a_n$ or $S \Rightarrow^* w$, i. e., $w \in L(G)$.

To fill the table, we work row-by-row upwards. Each row corresponds to one length of substrings; the bottom row is for strings of length 1, the second-from-bottom row for strings of length 2 and so on, until the top row corresponds to the one substring of length n which is w itself. The pseudocode is in Figure 1.

Genetic Programming

Genetic Programming (GP) is an evolutionary technique used to search over a huge state space of structured representations (computer programs). Each program represents a possible solution written in some language. The GP algorithm can be summarized in Figure 2 (Koza, 1992).

The evaluation of a solution is accomplished by using a set of training examples known as fitness cases which, in turn, is composed by sets of input and output data. Usually, the fitness is a measure of the deviation between the expected output for each input and the computed value given by GP (Banzhaf, Nordin, Keller & Francone, 2001).

Figure 1. The CYK algorithm

```
For r = 1 to n do
   V_r1 = { A | A → a_r ∈ P }

For s = 2 to n do
   For r = 1 to n-s+1 do
      V_rs = Ø
      For k = 1 to s-1 do
         V_rs = V_rs ∪ { A | A → BC ∈ P, B ∈ V_rk and C ∈ V_r+k, s-k
```

Figure 2. The GP algorithm

```
Generate Initial Population Randomly
While not (Stopping Condition)
   begin
      Evaluate the fitness of each individual
      Select individuals according to their fitness
      Modify them by applying genetic operators
   end
Returns the best individual found
```

There are two main selection methods used in GP: fitness proportionate and tournament selection. In the fitness proportionate selection, programs are selected randomly with probability proportional to its fitness. In the tournament selection, a fixed number of programs are taken randomly from the population and the one with the best fitness in this group is chosen. In this work, we use the tournament selection.

Reproduction is a genetic operator that simply copies a program to the next generation. Crossover, on the other hand, combines parts of two individuals to create two new ones. Mutation changes randomly a small part of an individual.

Each run of the main loop of GP creates a new generation of computer programs that substitutes the previous one. The evolution is stopped when a satisfactory solution is achieved or a predefined maximum number of generations is reached.

A GRAMMAR GENETIC PROGRAMMING APPROACH

We present how a GP approach can be applied to the inference of context-free grammars. First, we discuss the representation of the grammars. The modification needed in the genetic operators are also presented. In the last section, the grammar evaluation are discussed.

Initial Population

It is possible to represent a CFG as a list of structured trees. Each tree represents a production with its left-hand side as a root and the derivations as leaves. Figure 3 shows the grammar $G = (N, \Sigma, P, S)$ with $\Sigma = \{a, b\}$, $N = \{S, A\}$ and $P = \{S \rightarrow AS ; S \rightarrow b; A \rightarrow SA ; A \rightarrow a \}$.

The initial population can be created with random productions, provided that all the productions are reachable direct or indirectly starting with S.

Genetic Operators

The crossover operator is applied over a pair of grammars and works as follows. First, a production is chosen using a tournament selection. If the second grammar has no production with the same left-hand side of the production chosen, crossover is rejected. Otherwise, the productions are swapped.

The mutation operation is applied to a single selected grammar. A production is then chosen using the same mechanism of crossover. A new production, with the same left-hand side and with a randomly right-side, replaces the production chosen.

The crossover probability is usually high ($\approx 90\%$) and the mutation probability is usually low ($\approx 10\%$).

Unfortunately, using only the genetic operators mentioned, the convergence of the algorithm is not guaranteed. In our recently work, we demonstrated that the use of two local search operators is needed: an incremental learning operator (Rodrigues & Lopes, 2006) and an expansion operator (Rodrigues & Lopes, 2007). The first uses the information obtained from a CYK table to discover which production is missing to cover the sentence. The latter can expand the set of productions dynamically providing diversity.

Figure 3. An example of a CFG represented as a list of structured trees
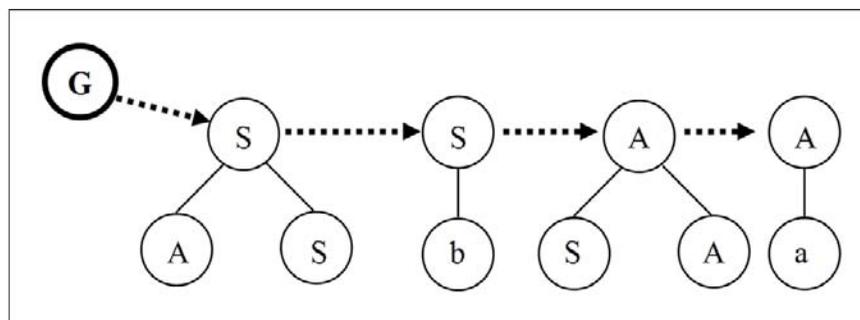
The Incremental Learning Operator

This operator is applied before the evaluation of each grammar in the population. It uses the CYK table obtained from the parsing of positive examples to allow the creation of an useful new production. The pseudocode is in Figure 4.

Once this process is completed with success, hopefully, there will be a set of positive examples (possible all) recognized by the grammar. Although, there is no warranty that some negative examples will still remain being rejected by the grammar.

The Expansion Operator

This operator adds a new nonterminal to the grammar and generates a new production with this new nonterminal as a left-side. This new approach allows grammars to grow dynamically in size. To avoid a new useless production, a production with another non-terminal in the left-side and the new non-terminal in the right-side is generated. It is important to emphasize that the new operator adds two productions to the grammar.

This operator promotes diversity in the population that is required in the beginning of the evolutionary process.

Grammar Evaluation

In grammatical inference, we need to train the system with both positive and negative examples to avoid overgeneralization. Usually the evaluation is done counting the positive examples covered by a grammar in proportion to the total of positive examples. If the grammar cover some negative examples, it is penalized in some way.

In our recently work, we use a **confusion matrix** that is typically used in supervised learning (Witten & Frank, 2005). Each column of the matrix represents the number of instances predicted either positively or negatively, while each row represents real classification of the instances. The entries in the confusion matrix have the following meaning in the context of our study:

- TP is the number of positive instances recognized by the grammar.
- TN is the number of negative instances rejected by the grammar.
- FP is the number of negative instances recognized by the grammar.
- FN is the number of positive instances rejected by the grammar.

Figure 4. The incremental learning operator pseudocode

```
For each positive example
    Construct the CYK table with V_rs
    If the example is not recognized
        If V_1n is not empty
            Then clone the root production changing
                 the left-hand side by S.
        Else
            If V_2,n-1 is not empty
                Then add production S → V_11 V_2,n-1
            Else
                For s = n-1 to n/2 do
                begin
                    If V_1s is not empty
                        Then If V_s+1, n-s is not empty
                            Then add production S → V_1s V_s+1,n-s
                end
```

There are a several measures that can be obtained from the confusion matrix. The most common is total accuracy that is obtained from the total of correct classified examples divided by the total number of instances. In this paper we used two other measures: specificity (Equation 1) and sensitivity (Equation 2). These measures evaluate how positive and negative examples are correctly recognized by the classifier.

$$\text{specificity} = \frac{TN}{TN + FP} \qquad (1)$$

$$\text{sensitivity} = \frac{TP}{TP + FN} \qquad (2)$$

The fitness is computed by the product of these measures leading to a balanced heuristic. This fitness measure was proposed by (Lopes, Coutinho & Lima, 1998) and widely used in many classification problems.

The use of confusion matrix provides a better evaluation of the grammars in the population, because grammars with the same accuracy rate usually has different values for specificity and sensitivity.

FUTURE TRENDS

The GP approach for the grammatical inference is based on the CYK algorithm and the confusion matrix. The preliminary results are promising but there are two problems that must be addressed.

The first is the that the solution found is not necessarily the smallest one. Depending on the run, the grammar inferred varies in size and, sometimes, it can be difficult to understand and may have useless or redundant production rules. Further work will focus on devising a mechanism able to favor shorter partial solutions.

The second is called "bloat", the uncontrolled growth of the size of an individual in the population (Monsieurs & Flerackers, 2001). The use of an expansion operator may cause this undesirable behavior. Nevertheless, this behavior was not detected in the experiments because all useless productions are eliminated during the search.

CONCLUSION

This article proposes a GP approach for context-free grammar inference. In this approach, an individual is a list of structured trees representing their productions with their left-hand side as the root and the derivations as leaves. It uses a local search operator, named Incremental Learning, capable of adjusting each grammar according to the positive examples. It also uses an expansion operator which adds a new production to the grammar allowing the grammars to grow in size. This operator promotes diversity in the population that is required in the earlier generations.

The use of a local search mechanism that is capable of learning from examples promotes a fast convergence. The preliminary results demonstrated that the approach is promising.

REFERENCES

Banzhaf, W., Nordin, P., Keller, R.E. & Francone, F.D. (2001) Genetic Programming: an introduction. San Francisco: Morgan Kaufmann.

Chomsky, N. (1957) Syntactic Structures. Paris: Mouton.

de la Higuera, C. (2005) A bibliographical study of grammatical inference. Pattern Recognition. 38(9), 1332-1348.

Dunay, B.D. (1994) Context free language induction with genetic programming. Sixth International Conference on Tools with Artificial Intelligence (ICTAI '94), IEEE Computer Society, 828-831.

Gold, E. M. (1967). Language identification in the limit. Information and Control. 10(5), 447-474.

Hopcroft, J. E., Motwani, R. & Ullman, J. D. (2001) Introduction to Automata Theory, Languages, and Computation. Addison-Wesley.

Huijsen, W. O. (1994) Genetic grammatical inference. CLIN IV. Papers from the Fourth CLIN Meeting, 59–72.

Javed, F. Bryant, B., Crepinsek, M., Mernik, M. & Sprague, A. (2004) Context-free grammar induction

using genetic programming. Proceedings of the 42nd. Annual ACM Southeast Conference '04, 404-405.

Korkmaz, E.E. & Ucoluk, G. (2001) Genetic programming for grammar induction. Proceedings of 2001 Genetic and Evolutionary Computation Conference Late Breaking Papers, 245-251.

Koza, J.R. (1992) Genetic Programming: On the Programming of Computers by Natural Selection. Cambridge: MIT Press.

Lopes, H.S., Coutinho, M.S. & Lima, W.C. (1998) An evolutionary approach to simulate cognitive feedback learning in medical domain. Proceedings of the Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives, 193-207.

Monsieurs, P. & Flerackers, E. (2001) Reducing bloat in genetic programming. Proceedings of the International Conference, 7th Fuzzy Days on Computational Intelligence, Theory and Application, LNCS v. 2206, Springer-Verlaq, 471-478.

Rodrigues, E. & Lopes, H.S. (2007) Genetic Programming for induction of context-free grammars. Proceedings of the 7th. International Conference on Intelligent Systems Design and Applications (ISDA'07) (*to be published*).

Rodrigues, E. & Lopes, H.S. (2006) Genetic programming with incremental learning for grammatical inference. Proceedings of the 6th. International Conference on Hybrid Intelligent Systems (HIS'06), IEEE Press, Auckland, 47.

Sen, S. & Janakiraman, J. (1992) Learning to construct pushdown automata for accepting deterministic context-free languages. Proceedings of the Applications of Artificial Intelligence X: Knowledge-Based Systems, SPIE v. 1707, 207–213.

Starckie, B,. Costie, F. & van Zaanen, M. (2004) The Omphalos context-free grammar learning competition. Proceedings of the International Colloquium on Grammatical Inference, 16-27.

Witten, I. H. & Frank, E. (2005) Data Mining. San Francisco: Morgan Kaufmann.

Wyard, P. (1994) Representational issues for context free grammar induction using genetic algorithms. Proceedings of the Second International Colloquium in Grammatical Inference (ICGI-94). LNAI n 862. Springer-Verlag, 222–235.

Wyard, P. (1991) Context free grammar induction using genetic algorithms. Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA'91). Morgan Kaufmann, 514-518.

KEY TERMS

CYK: A Cocke-Younger-Kasami algorithm used to determine whether the sentence can be generated by the grammar.

Evolutionary Computation: Large and diverse class of population-based search algorithms that is inspired by the process of biological evolution through selection, mutation and recombination. They are iterative algorithms that start with an initial population of candidate solutions and then repeatedly apply a series of the genetic operators.

Finite Automata: A model of behavior composed of a finite number of states, transitions between those states, and actions. They are used to recognize regular languages.

Genetic Algorithm: A type of evolutionary computation algorithm in which candidate solutions are represented typically by vectors of integers or bit strings, that is, by vectors of binary values 0 and 1.

Heuristic: Function used for making certain decisions within an algorithm; in the context of search algorithms, typically used for guiding the search process.

Local Search: A type of search method that starts at some point in search space and iteratively moves from position to neighbouring position using heuristics.

Pushdown Automata: A finite automaton that can make use of a stack containing data. They are used to recognize context-free language.

Search Space: Set of all candidate solutions of a given problem instance.

Evolutionary Robotics

J. A. Becerra
Universidade da Coruña, Spain

R. J. Duro
Universidade da Coruña, Spain

INTRODUCTION

Evolutionary Robotics is a field of **Autonomous Robotics** where the controllers that implement behaviours are obtained through some kind of Evolutionary Algorithm. The aim behind this technique is to obtain controllers minimizing human intervention. This is very interesting in order to achieve complex behaviours without introducing a "human bias". Sensors, body and actuators are usually different for a human being and for a robot, so it is reasonable to think that the best strategy obtained by the human designer is not necessarily the best one for the robot. This article will briefly describe Evolutionary Robotics and its advantages over other approaches to Autonomous Robotics as well as its problems and drawbacks.

BACKGROUND

The firsts modern attempts to obtain a **robot** that could be called "autonomous", that is, with the ability of adapting to a non predefined environment and perform its tasks adequately, are from the late sixties and they basically tried to reproduce human reasoning in the robot. The reasoning process was divided into several steps (input data interpretation, environment modelling, planning and execution) that were performed sequentially. As time passed, robots were getting better thanks to better design and construction, more computational capabilities and improvements in the Artificial Intelligence techniques employed. But also some problems appeared and remained there: lack of reaction in real time, inability to handle dynamic environments and unmanaged complexity as desired behaviours become more complex.

In the late eighties a new approach, called **Behaviour Based Robotics**, was introduced. It emphasized the behaviour, no matter how it was obtained, as op-posed to traditional (knowledge based) Autonomous Robotics where the emphasis was on modelling the knowledge needed to perform the behaviour. This new approach proposes a direct connection between sensors and actuators with no explicit environment modelling. Behaviour Based Robotics has proven to be very useful when implementing low level behaviours, but it has also shown problems when scaling to more complex behaviours. Phil Husbands (Phil Husbands et al., 1994) and Dave Cliff (Cliff et al., 1993a) have shown that it is not easy to design a system that connects sensors and actuators in order to achieve complex behaviours. Regardless of whether the system is monolithic (to design a complex system in just one step is never easy) or modular the design problem is difficult basically due to the fact that the possible interactions between modules grow exponentially. An additional problem is that human designed controllers for autonomous robots are not necessarily the best choice, sometimes they are simply not a good choice. A human designer cannot avoid perceiving the world with its own sensors and developing solutions for problems taking into account the perceptions and the actuations he / she can perform. Furthermore, humans tend to simplify and modularize problems and this is not always possible in complex environments.

Due to these drawbacks, in the early nineties some researchers started to use Evolutionary Algorithms in order to automatically obtain controllers for autonomous robots leading to a new robotics field: Evolutionary Robotics. Some examples of these research line are the papers by Irman Harvey (Harvey et al., 1993), Phil Husbands (Husbands et al, 1994), Dave Cliff (Cliff et al., 1993a) and Randall Beer and John Gallagher (Randall Beer and John Gallagher, 1992). The idea is very simple and very promising and, again, has shown it is very effective with simple behaviours. But, even if it solves some problems, it also has its own problems when dealing with complex behaviours. In the next sec-

tion we will talk about those problems and, in general, about the main aspects to take into account when using evolution in Autonomous Robotics.

EVOLUTIONARY ROBOTICS

The basis of Evolutionary Robotics is to use **evolutionary algorithms** to automatically obtain robot controllers. In order to do that, there are many decisions to be made. First of all, one must decide what to evolve (controllers, morphology, both?). Then, whatever is to be evolved has to be encoded in chromosomes. An evolutionary algorithm must be chosen. It has to be decided where and how to evaluate each individual, etc. These issues will be addressed in the following sections.

What to Evolve

The first decision to be made is what to evolve. The most common choice is to evolve controllers for a given robot, but we can also evolve the morphology or both things together. If we choose to evolve only the controllers, we also have to decide how they will be implemented. The most usual choices are artificial neural networks, fuzzy logic systems and classifier systems.

Classifier systems are made up of rules (the classifier set). Each rule consists of a set of conditions and a message. If the conditions are accomplished, a message can produce an action on an actuator and is stored in a message list. Sensor values are also stored in this message list. Messages in the message list may change the state of conditions, leading to a different set of activated rules. There is an apportionment of credit system that changes the strength for each rule and a rule discovery system, where a genetic algorithm generates new rules using existing rules in the classifier set and their strength. An example of classifier systems is the work of Dorigo and Colombetti (Colombetti et al, 1996), (Dorigo and Colombetti, 1993, 1995, 1998).

Fuzzy logic has also been used to encode controllers. Possible sensed values and acting values are encoded into predefined fuzzy sets and the rules that relate both things can be evolved. Examples: (Cooper, 1995), (Hoffmann and Pfister, 1994), (Vicente Matellán et al, 1998).

Artificial neural networks are the most common way of implementing controllers in evolutionary robotics.

On one hand, they are noise and failure tolerant and, on the other, they can be used as universal function approximators and can be easily integrated with an evolutionary algorithm to obtain a controller from scratch. Many researchers have used ANNs, just to mention some of them: (Beer and Gallagher, 1992), (Cliff et al, 1992), (Floreano and Mondada, 1998), (Harvey et al, 1993), (Kodjabachian and Meyer, 1995), (Lund and Hallam, 1996), (Nolfi et al, 1994) and (Santos and Duro, 1998).

How to Encode What We are Evolving

When encoding a controller into the chromosome, the most obvious choice, and the most common one, is to make a direct encoding. That is, each controller parameter becomes a gene in the chromosome. For instance, if the controller is an ANN, each synaptic weight as well as the biases and other possible parameters that describe the ANN topology correspond to a gene, (Mataric and Cliff, 1996), (Miglino et al, 1995a). This can lead to very large chromosomes, as the chromosome size grows proportional to the square of the network size (in case of feedforward networks), increasing the dimensionality of the search space and making it more difficult to obtain a solution in reasonable time. Another problem is that the designer has to predefine the full topology (size, number of neurons, etc.) of the ANN, which is, in general, not obvious usually leading to a trial and error procedure. To address this problem, some researchers employ encoding schemes where the chromosome length may vary in time (Cliff et al., 1993b).

Another possibility is to encode elements that, following a set of rules, encode the development of the individual (Guillot and Meyer, 1997), (Angelo Cangelosi et al, 1994), (Kodjabachian and Meyer, 1998). Some authors even simultaneously evolve with this system both the controller and the morphology, but mostly for virtual organisms (Sims, 1994) or very simplified real robots.

Where to Carry Out the Evolution Process

To determine how good an individual is, it is necessary to evaluate this individual in an environment during a given time interval. This **evaluation** has to be performed more than once in order to make the process indepen-

E

dent from the initial conditions. The more complex the behaviour is the more time that is required to evaluate an individual. The evaluation of the individual is usually the most time consuming phase, by far, in the whole evolutionary process for evolutionary robotics. The evaluation can be carried out in a real environment, in a simulated environment or both.

Evaluation in a real environment has the obvious advantage that the controllers obtained will work without problems in the real robot. But it is much slower than evaluation in simulated environments, it presents the danger of harming the robot and many limitations on the evaluation functions that may be used. These is why researchers that consider evaluation in a real environment mostly use small robots in controlled environments (Dario Floreano and Francesco Mondada, 1995, 1996, 1998).

The alternative is to perform the evaluation in a simulated environment (Beer and Gallagher, 1992), (Cliff et al, 1993a), (Meeden, 1996), (Miglino et al, 1995a, b). This is faster and it permits parallelizing the algorithm and using fitness functions that are impossible in a real environment. Nevertheless, it has the additional problem of how to carry out this simulation in order to obtain controllers that work in the real world. Jakobi (Jakobi 1997) formalized this problem and established the conditions to be taken into account in order to successfully transfer the controllers obtained in simulation to the real robot.

Some researchers choose to perform a simulated evaluation for the different generations of the evolutionary process except in the last generations where a real evaluation is performed to make sure that the controllers work in the real robot. Nevertheless, this approach presents the same problems, although somehow reduced, as the case of evolution in a real environment (Miglino et al, 1995a).

How to Evaluate

Once the previous choices have been made, it is necessary to decide how the **fitness** will be calculated. There are two different perspectives for doing this: a local perspective or a global perspective (Mondada and Floreano, 1995). The first one consists in establishing for each step of the robot life a fitness of its actions in relation to its goal. The final fitness will be the sum of the fitness values corresponding to each step. This strategy presents two main drawbacks. Except in toy problems,

it is very difficult to decide beforehand the fitness of each action towards a final objective. Sometimes the same action may be good or bad depending on what happened before or after in the context. In addition, this approach implies an external determination of goodness as it is the designer who is imposing through these action-fitness pairs how the robot must act.

The global approach, on the other hand, implies defining a fitness criteria based on how good the robot was at achieving its final goal. The designer does not specify how good each action is in order to maximize fitness, and the evolutionary algorithm has a lot more freedom for discovering the best final controller. The main problem here is that there is a lot less knowledge injected in the evolutionary process, and thus the evolution may find loopholes in the problem specification and thus maximize fitness without really achieving the function we seek.

There are two main ways in which global fitness can be obtained: external and internal. By external we mean a fitness assigned by someone or some process outside the robot. An extreme example of external global evaluation of the robot behaviour is presented in (Lund et al,1998).

The other possible approach is to employ an internal representation of fitness. This is, employ clues in the environment the robot is conscious of and that it can use in order to judge its level of fitness without the help from any external evaluator (apart from the environment itself). A concept often used in order to implement this approach is that of internal energy. The robot has an internal energy level and this energy level increases or decreases according to a set of rules or functions related to robot perceptions. The final fitness of the robot is given by the level of energy at the end of its life.

FUTURE TRENDS

The main problem of the evolutionary approach to robotics is that, although it is easy to obtain simple behaviours, it does not scale well to really complex behaviours without introducing some human knowledge in the system. An obvious solution to this problem is the reduction of evaluation time, which is slowly happening thanks to the increasing computational capabilities. Another way of controlling this problem is to try to obtain a better encoding scheme using development.

Another problem that arises when trying to minimize human intervention is that the evolutionary algorithm can lead to a behaviour that optimizes the defined fitness function but the result does not correspond with the desired behaviour due to an incomplete or inadequate fitness function. This problem must be addressed from a theoretical and formal point of view so that the appropriate fitness functions can be methodologically obtained.

Finally, there is a problem in common with Autonomous Robotics: **benchmarking**. It is not easy to compare different approaches and usually we can only say if a behaviour is satisfactory or not. It is hard to compare it with other similar behaviours or to know if the results would be better by changing something in the approach followed.

CONCLUSION

Evolutionary Robotics has quickly developed from its birth in the beginning of the nineties as the most common way of obtaining behaviours in Behaviour Based Robotics. It has shown that it is the easiest way to automatically obtain controllers for autonomous robots while trying to minimize the human factor, at least when the behaviours are not too complex, due to the fact that it is easy to encode every tool used to implement controllers and obtain a solution by just defining a fitness function. Inside Evolutionary Robotics, simulated evaluation is the preferred way of evaluating how good a candidate solution is and Artificial Neural Networks are the preferred tool to implement controllers due to their noise and failure tolerance and their nature as a universal function approximator. Due to the time required to evaluate individuals and the number of individuals that must be evaluated, the selection of the correct evolutionary algorithm and its parallelization are critical factors.

REFERENCES

Beer, R.D. and Gallagher, J.C. (1992), "Evolving Dynamical Neural Networks for Adaptive Behavior", Adaptive Behavior, Vol. 1, No. 1, pp. 91–122.

Cangelosi, A., Parisi, D., and Nolfi, S. (1994), "Cell Division and Migration in a Genotype for Neural Networks", Network, Vol. 5, pp. 497–515.

Cliff, D., Harvey, I., and Husbands, P. (1992), Incremental Evolution of Neural Network Architectures for Adaptive Behaviour, Tech. Rep. No. CSRP256, Brighton, School of Cognitive and Computing Sciences, University of Sussex, UK.

Cliff, D., Harvey, I., and Husbands, P. (1993a), "Explorations in Evolutionary Robotics", Adaptive Behavior, Vol. 2, pp. 73–110.

Cliff, D., Husbands, P. and Harvey, I. (1993b), "Evolving Visually Guided Robots", From Animals to Animats 2, Proceedings of the Second International Conference on Simulation of Adaptive Behaviour (SAP 92), J–A. Meyer, H. Roitblat and S. Wilson (Eds.), MIT Press Bradford Books, Cambridge, MA, pp. 374–383.

Colombetti, M., Dorigo, M., and Borghi, G. (1996), "Behavior Analysis and Training – A Methodology for Behavior Engineering", IEEE Transactions on Systems, Man and Cybernetics, Vol. 26, No. 3, pp. 365–380.

Cooper, M.G. (1995), "Evolving a Rule–Based Fuzzy Controller", Simulation, Vol. 65, No. 1.

Dorigo, M. and Colombetti, M. (1993), "Robot Shaping: Developing Autonomous Agents through Learning", Artificial Intelligence, Vol. 71, pp. 321–370.

Dorigo, M. (1995), "ALECSYS and the AutonoMouse: Learning to Control a Real Robot by Distributed Classifier Systems", Machine Learning Journal, Vol. 19, No. 3, pp. 209–240.

Dorigo, M. and Colombetti, M. (1998), Robot Shaping: An Experiment in Behavior Engineering, MIT Press.

Floreano, D. and Mondada, F. (1996), "Evolution of Homing Navigation in a Real Mobile Robot", IEEE Transactions on Systems, Man, and Cybernetics, Part–B, Vol. 26, pp. 396–407.

Floreano, D. and Mondada, F. (1998), "Evolutionary Neurocontrollers for Autonomous Mobile Robots", Neural Networks, Vol. 11, pp. 1461–1478.

Guillot, A. and Meyer, J–A. (1997), "Synthetic Animals in Synthetic Worlds", In Kunii et Luciani (Eds), Synthetic Worlds, John Wiley and Sons.

Harvey, I., Husbands, P., and Cliff, D. (1993), "Issues in Evolutionary Robotics", From Animals to Animats 2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92), J–A.

Meyer, H. Roitblat, and S. Wilson (Eds.), MIT Press, Cambridge, MA, pp. 364–373.

Hoffmann, F. and Pfister, G. (1994), "Automatic Design of Hierarchical Fuzzy Controllers Using Genetic Algorithms", Proceedings of EUFIT 94, Aachen, Germany.

Husbands, P., Harvey, I., Cliff, D. and Miller, G. (1994), "The Use of Genetic Algorithms for the Development of Sensorimotor Control Systems", P. Gaussier and J–D. Nicoud (Eds.), From Perception to Action, IEEE Computer Society Press, Los Alamitos CA, pp. 110–121.

Jakobi, N., (1997), "Half Baked, Ad Hoc and Noisy Minimal Simulations for Evolutionary Robotics", Fourth European Conference on Artificial Life, P. Husbands and I. Harvey (Eds.), MIT Press, pp. 247–269.

Kodjabachian, J. and Meyer, J–A (1995), "Evolution and Development of Control Architectures in Animats", Robotics and Autonomous Systems, Vol. 16, pp. 161–182.

Kodjabachian, J. and Meyer, J–A (1998), "Evolution and Development of Modular Control Architectures for 1–D Locomotion in Six–Legged Animats", Connection Science, Vol. 10, No. 3–4, pp. 211–37.

Lund, H.H. and Hallam, J.C. (1996), "Sufficient Neurocontrollers can Be Surprisingly Simple", Research Paper 824, Department of Artificial Intelligence, University of Edinburg.

Lund, H.H., Miglino, O., Pagliarini, L., Billard, A., and Ijspeert, A. (1998), "Evolutionary Robotics – A Children's Game", Proceedings of IEEE 5th International Conference on Evolutionary Computation.

Mataric, M.J. and Cliff, D. (1996), "Challenges in Evolving Controllers for Physical Robots", Robotics and Autonomous Systems, Vol. 19, No. 1, pp. 67–83.

Matellán, V., Fernández, C., and Molina, J.M. (1998), "Genetic Learning of Fuzzy Reactive Controllers", Robotics and Autonomous Systems, Vol. 25, pp. 33–41.

Meeden, L. (1996), "An Incremental Approach to Developing Intelligent Neural Network Controllers for Robots", IEEE Transactions on Systems, Man and Cybernetics Part. B: Cybernetics, Vol. 26, No. 3, pp. 474–485.

Miglino, O., Lund, H.H., and Nolfi, S. (1995a), "Evolving Mobile Robots in Simulated and Real Environments", Artificial Life, Vol. 2, No. 4, pp. 417–434.

Miglino, O., Nafasi, K., and Taylor, C. (1995b), "Selection for Wandering Behavior in a Small Robot", Artificial Life, Vol. 2, pp. 101–116.

Mondada, F. and Floreano, D. (1995), "Evolution of Neural Control Structures: Some Experiments on Mobile Robots", Robotics and Autonomous Systems, Vol. 16, pp. 183-195.

Nolfi, S., Elman, J., and Parisi, D. (1994), "Learning and Evolution in Neural Networks", Adaptive Behavior, Vol. 1, pp. 5–28.

Santos, J. and Duro, R.J. (1998), "Evolving Neural Controllers for Temporally Dependent Behaviors in Autonomous Robots", Tasks and Methods in Applied Artificial Intelligence, A.P. del Pobil, J. Mira and M. Ale (Eds.), Lecture Notes in Artificial Intelligence, Vol. 1416, Springer–Verlag, Berlín, pp. 319–328.

Sims, K. (1994), "Evolving 3D Morphology and Behavior by Competition", R. Brooks and P. Maes (Eds.), Alife IV, MIT Press, Cambridge, MA, pp. 28–39.

KEY TERMS

Artificial Neural Network: An interconnected group of artificial neurons, which are elements that use a mathematical model that reproduce, through a great simplification, the behaviour of a real neuron, used for distributed information processing. They are inspired by nature in order to achieve some characteristics presented in the real neural networks, such as error and noise tolerance, generalization capabilities, etc.

Autonomous Robotics: The field of Robotics that tries to obtain controllers for robots so that they are tolerant and may adapt to changes in the environment.

Behaviour Based Robotics: The field of Autonomous Robotics that proposes not to pay attention to the knowledge that leads to behaviours, but just to implement them somehow. It also proposes a direct connection between effectors and actuators for every controller running in the robot, eliminating the typical sensor interpretation, world modelling and planning stages.

Evolutionary Algorithm: Stochastic population based **s**earch algorithm inspired on natural evolution. The problem is encoded in an n-dimensional search space where individuals represent candidate solutions. Better individuals have higher reproduction probabilities than worse individuals, thus allowing the fitness of the population to increase through the generations.

Evolutionary Robotics: The field of Autonomous Robotics, usually also considered as a field of Behaviour Based Robotics, that obtains the controllers using some kind of evolutionary algorithm.

Knowledge Based Robotics: The field of Autonomous Robotics that tries to achieve "intelligent" behaviours through the modelling of the environment and a process of planning over that model, that is, modelling the knowledge that generates the behaviour.

MacroEvolutionary Algorithm: Evolutionary algorithm using the concept of species instead of individuals. Thus, low fitness species become extinct and new species appear to fill their place. Its evolution is smoother, slower but less inclined to fall into local optima as compared to other evolutionary algorithms.

Evolved Synthesis of Digital Circuits

Laurenţiu Ionescu
University of Pitesti, Romania

Alin Mazare
University of Pitesti, Romania

Gheorghe Şerban
University of Pitesti, Romania

Emil Sofron
University of Pitesti, Romania

INTRODUCTION

Traditionally physical systems have been designed by engineers using complex collections of rules and principles. The design process is top-down in nature and begins with a precise specification. This contrasts very strongly with the mechanisms which have produced the extraordinary diversity and sophistication of living creatures. In this case the ''designs'' are evolved by a process of natural selection. The design starts as a set of instructions encoded in the DNA whose coding regions are first transcribed into RNA in the cell nucleus and then later translated into proteins in the cell cytoplasm. The DNA carries the instructions for building molecules using sequences of amino acids. Eventually after a number of extraordinarily complex and subtle biochemical reactions an entire living organism is created. The survivability of the organism can be seen as a process of assembling a larger system from a number of component parts and then testing the organism in the environment in which it finds itself (Miller, 2000).

The main target of the **evolvable hardware** is to build a digital circuit using bio inspired methods like genetic algorithms. Here the potential solutions are coded like configuration vectors which command interconnection between logical cells inside the reconfigurable circuit. All configuration vectors represent the genotype and one single configuration vector is the individual with its own characteristics (like chromosome).

The individuals are generated by genetic operators like crossover or mutation. One individual give one solution circuit which is tested in **evaluation** module. The circuit obtained from the individual consist the phenotype. The circuit behavior is compared with target functions, which we desire to implement. The result is fitness: if the circuit approximates the behavior of the target function, we have a good fitness for the individual which generate the circuit. Then each individual whit its fitness gets into selection module where the future parents in crossover and mutation are decided. Finally we have a circuit solution which implements the target function. We have an evolved synthesis of digital circuit – a method like assemble and test.

This method can be useful because explore the design space beyond the limits imposed by traditional design methods. Two research directions are developed in evolvable hardware. In extrinsic evolvable hardware the individuals are obtained from software implementation on computer and phenotype consist in high level abstract circuits like SPICE object files or FPGA configuration files (.bit). The **intrinsic evolution**, on the other hand, supposes that entire evolution process is inside one or more chips (FPGA): the hardware implementation of evolved hardware.

The challenge is to design an intrinsic evolution because can be used for applications like robots control system. But this involves implementation of the software based algorithms in hardware modules.

BACKGROUND

The **dynamic reconfigurable hardware** area and evolvable hardware knows, in the last years, a fast evolution. Ten years ago the digital circuit implementation, with high degree of complexity involve more problems caused specially by technologies limits. The

market was up most by complex programmable gates array or by the low grains field programmable gates array where upon the main problems are the number of Boolean cells available on chip and the delay time. The fast evolution of the technologies increases in our day the performance of programmable circuits. Thus, today is possible to implement a high speed central processing units core which is comparable with the application specific integrated circuit implementations. Therewith the low product costs make that a modern programmable digital circuit can be purchased by end users like students and researchers. Thus an evolution in designing, synthesis and implementation techniques with programmable logic circuits is required. One very attractive direction of research is implementation of hardware bio-inspired systems on programmable logic circuits like neural networks or evolutionary algorithms (e.g. genetic algorithms).

The first research direction in this area is to find solutions for improve the genetic algorithm performance by hardware implementation. Software implementations have the advantage of flexibility and easily configuration. However, the convergence speed is slow because the serial execution of the steps whiles the algorithm run. To increase the speed a parallel implementation of the modules is required (Goldberg, 1995). This is done by hardware implementation on programmable logic circuits. More investigations are done in this area.

The second research direction is to join the concept of assemble-and-test together with an evolutionary algorithm to gradually improve the quality of a design has largely been adopted in the nascent field of **evolvable hardware** where the task is to build an electronic circuit.

Thompson (Thompson, 1999) makes the first research, uses a reconfigurable platform and showed that is possible to evolve a circuit which could discriminate between two square wave signals. He demonstrates that is possible to design digital circuit using evolved algorithm. The evolutionary process had utilized physical proprieties of the underlying silicon substrate to produce an efficient circuit. He argued that artificial evolution can produce design for electronics circuit which lies outside the scope of conventional methods. Koza (Koza, 1997) have pioneered the extrinsic evolution of analogue circuit using SPICE and have automatically generated circuit which are competitive with human designer.

Most workers are content with the extrinsic evolution because the evolutionary algorithm is software – based. But Scott and others achieve different solutions for hardware implementation of evolutionary algorithms. In (Scott, 1997) is design a **hardware genetic algorithm** implemented as pipe line hardware structure in FPGA. His work is a demonstration that full integrated **evolved hardware** (**intrinsic**) solution can be implemented. To design hardware modules for crossover, selection and mutation are used **combinational network**s such as systolic arrays presented by (Bland2001).

Miller et al. (Miller,2000) give a reference with his work concerned of the evolution of combinational digital circuit to implement arithmetic functions. First he uses the gates networks which evolve in arithmetic circuit but demonstrate that is possible to use evolution of some sub-circuits to achieve more complex circuits.

Another example of extrinsic evolved synthesis is give by Martin's work (Martin 2001). Here the phenotype is give by a hardware description language (HandleC) sequences.

But utilization of genetic algorithm in hardware design is in any more areas. In (Shaaban 2001) is used in integrated circuit design in semiconductor detectors. Yasunaga (Yasunaga 2001) use a hardware synthesis of digital circuit in reconfigurable programmable logic array structure to implement speach recognition system. Recently evovable synthesis is used for sequential circuit design(Ali 2004) or in digital filters design (Iana2006).

The evolved combinational and sequential synthesis for digital circuits is included as control module for self-contained mobile system to execute more tasks like obstacle avoidance and target hit in (Sharabi2006). In fact the solving of the multi objective (multi task) problems in hardware system by evolutionary is treated by Coello (Collelo 2002) and used in more recent works (Zhao 2006).

The question is: is possible to design digital circuits using an evolutionary algorithm like genetic algorithm (GA)?

To answer of this question, first, the design of a reconfigurable circuit which can be programmed by evolutionary algorithm is required. A solution can be reconfigurable multilayer gates network. Each gate in layer x can be connected or not whit a gate in layer x+1.

RECONFIGURABLE GATES NETWORK AND HARDWARE GENETIC ALGORITHM

Models Used

This section is dedicated to elaborate models for the two components of the **evolvable hardware microstructure**s: **hardware genetic algorithm** and **dynamically reconfigurable circuit**.

In this section is presented first the concept of generic dynamically reconfigurable structures. Here each cell of the network is give by computing units and configurable network unit. To run the algorithm a local computation is perform - arithmetic and logic operation by each cell, and a network computation – each cell can configure the connection with the others cell in the network.

This concept can be extended to multilayer gates network by replace the local computations cell with elementary gates and network connections with switches commands by genetic algorithm (Ionescu2004). In this particularly case dynamically reconfigurable structure became a hardware reconfigurable structure.

In figure 1 is presented the cell of reconfigurable hardware: the generic digital gate.

Each **generic gate** can be configured with a local elementary Boolean function and more switch connections with another gates. Thus, the first coding schema for genetic algorithm is conceived: the individual is a string of connections status and Boolean functions code like Fig1.b.

Each algorithm contains functions which can be executed by central processing unit. To the compute systems with single processor one single function can be executed at time. In hardware structure, each function can be implemented in **combinational network**. There are two main kind of **combinational network**s: the sorting network composes by min/max elementary circuits and the permutation network composes by permutation elementary circuit. In figure 2 are presented the recursive sorting circuit and the recursive permutation circuit.

Figure 1. a) The generic gate – cell of the hardware reconfigurable structure, b) gates network coded as configuration vector
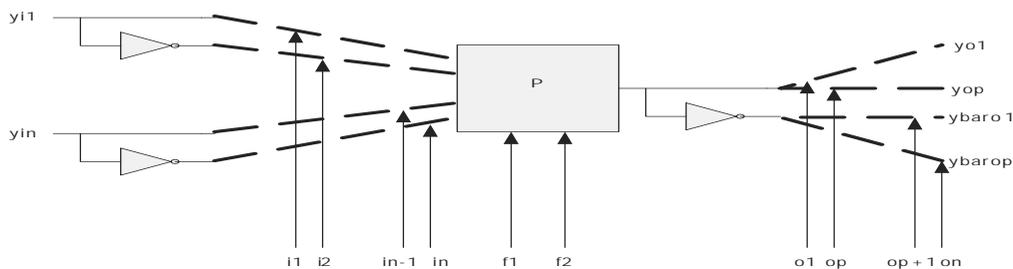
(a)

Gate 1			Gate 2			...	Gate n		
inputs	function	outputs	inputs	function	outputs	…	inputs	function	outputs

(b)

Figure 2. The recursive sorting and permutation circuit used for hardware implementation of the genetic algorithm modules
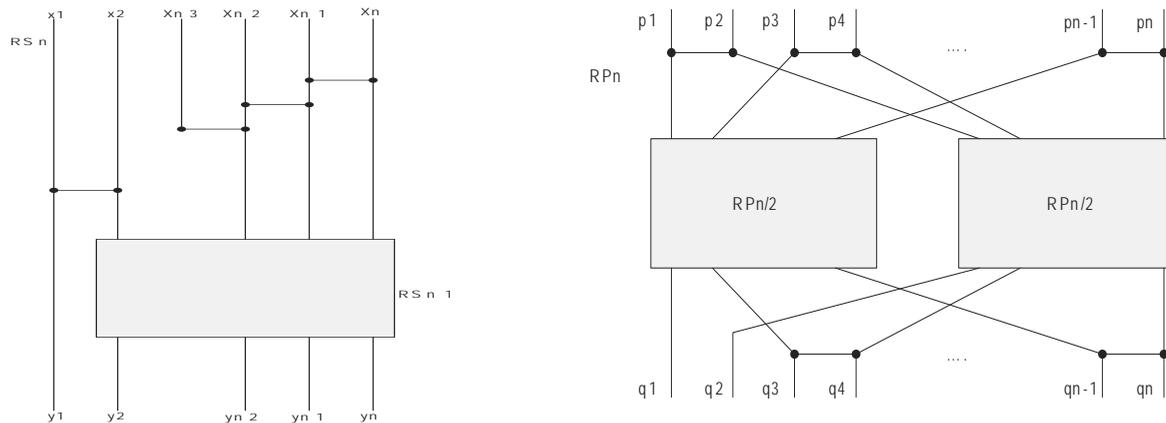
Design of Hardware Genetic Algorithm, Dynamic Reconfigurable Circuit and Application Description

This section gets into the techniques used for evolvable hardware **microstructure**s design. First it presents the design of hardware genetic algorithm by using the models from the preceding section. Each module are designed individually, describe with combinational networks. The hardware genetic algorithm is a circuit which connect all modules in a fully hardware solution. The block diagram of **HGA (hardware genetic algorithm)** is presented in figure below (fig.3).

The main issue of this solution is that the modules can be used in another algorithm and can be interconnected in another way without to be redesigned. Each module can work individual, therefore the structure can process more generation in the same time. On the other side, each module was designed as networks of elementary functions like min/max or permutation. Thus any new change claimed like increasing the size or number of individuals is very easily to made by adding the elementary functions blocks. In the selection module individual (bits string) does, with fitness computed, enter in the left side of the array. Each cell collates fitness values from two inputs x_{in} and y_{in}. The output x_{out} get the individual with the smallest fitness value from the inputs and the output y_{out} individual with

the biggest one. So, the individuals with poor (small value) fitness will cross array on horizontal, from the left to right and individuals with good fitness will cross array from top to bottom on vertical. Finally, we have in the left side outputs with the best fitness.

The same concept are used to crossover circuit. Some individual sorted by selection module enter in crossover module. Operator is applied on a certain number of pair of individuals. Usually, individuals with the best fitness are "parents" for generation 1 of offspring which result from this module.

For mutation module we use one single column of logical structures (mutation cells) too. Array is design with follow restriction: mutation must affect only one single bit from an individual and one single individual from generation (all individuals from one iteration step). Structure can be easily changed if we want another behavior.

Evaluation is done by comparing response of partial solutions provide by each individual and the desirable response. Here is defining a target Boolean function, function which must be implemented in hardware. Another **evaluation** criterion is give by minimization of digital resources used to implement target function. The fitness value is compute consulting both evaluation criterions.

Figure 3. Hardware genetic algorithm

SELECTION

$$x_{out} = \begin{cases} y_{in}\, , & x_{in} > y_{in} \\ x_{in}\, , & x_{in} \le y_{in} \end{cases}$$

$$y_{out} = \begin{cases} y_{in}\, , & x_{in} \le y_{in} \\ x_{in}\, , & x_{in} > y_{in} \end{cases}$$

Input strings (ind.)

Output strings (ind.)

Parents

CROSSOVER

RNG

Inputs strings (parents)

Output strings (offspring)

$$x_{out1} = \begin{cases} x_{in1}\, , & y_{in} < p \\ x_{in2}\, , & y_{in} \ge p \end{cases}$$

$$x_{out2} = \begin{cases} x_{in2}\, , & y_{in} < p \\ x_{in1}\, , & y_{in} \ge p \end{cases}$$

$$y_{out} = f(y_{in})$$

MUTATION

RNG1 R NG2

Inputs strings (parents)

Output strings (offspring)

$$B_{z_{in}}(x_{out}) = \begin{cases} B_{z_{in}}(x_{in})\, , & y_{in} \ne n \\ \overline{B}_{z_{in}}(x_{in})\, , & y_{in} = n \end{cases}$$

$$y_{out} = f(y_{in})$$
$$z_{out} = g(z_{in})$$

EVALUATION

target

fitness computation

individ 1

individ 2

individ n

interface

clk

Individuals and fitness

Generation with new individuals

Random numbers

PSEUDO-RANDOM NUMBER GENERATOR

613

$$f_{total} = 0.9\, f_{eval.corect} + 0.1\, f_{min\, in}$$

Three **dynamic reconfigurable circuits** are designed and tested. All are based on hardware reconfigurable structure presented in figure 1. First schema, min-max terms reconfigurable circuit, use the same principles as programmable logic array. The scheme is composed by three layers: INV layer, AND layer and OR layer. Genetic algorithm command connections between INV layer and AND layer. This reconfigurable circuit has the fast convergence speed and the individuals with the smallest size but explore only traditional space solution and its size grow exponentially with inputs number and linear with outputs number. The second circuit is reconfigurable INV-AND-OR circuit. Like the first circuit, it has three layers: INV layer, AND layer and OR layer. Genetic algorithm configure in this case connections between INV - AND layer and AND – OR layer. This schema reduces the increase of size with number of outputs but remain exponential increase with number of inputs and the size of individuals is bigger than first circuit.

The last reconfigurable circuit is elementary functions reconfigurable circuit (e – reconfigurable). It contains more layers. Each layer contains a number of

generic gates. **Generic gate** can implement a Boolean elementary functions (AND, OR, XOR) and more complex circuits like MUX. This solution increases the size of the individuals and the complexity of the reconfigurable circuit but is almost invariant with number of inputs and outputs.

The last reconfigurable circuit explores the largest solution space, beyond the bounds of traditional design methods.

The **evolvable hardware** is used in three applications. First, the target function is static and algorithm must find hardware solution to implement it. Each individual represent here a potential solution for hardware implementation of target function. Evolution loop is repeated until optimal solution is found. Hardware solution finding here is named evolved hardware. At this time evolution loop is stopped.

In the second application the target function is also static. But here the individual codes only a sub circuit – one **generic gate** or one gates layer. The individuals evolve and the offspring replace the parents in different position and **evaluation** is done to entire circuit until the new solution is better than the old solution. Evolution loop is repeated until optimal solution is found.

This solution is used to design circuit with big number of inputs, outputs and sub circuits.

Figure 4. Reconfigurable elementary functions circuit

E
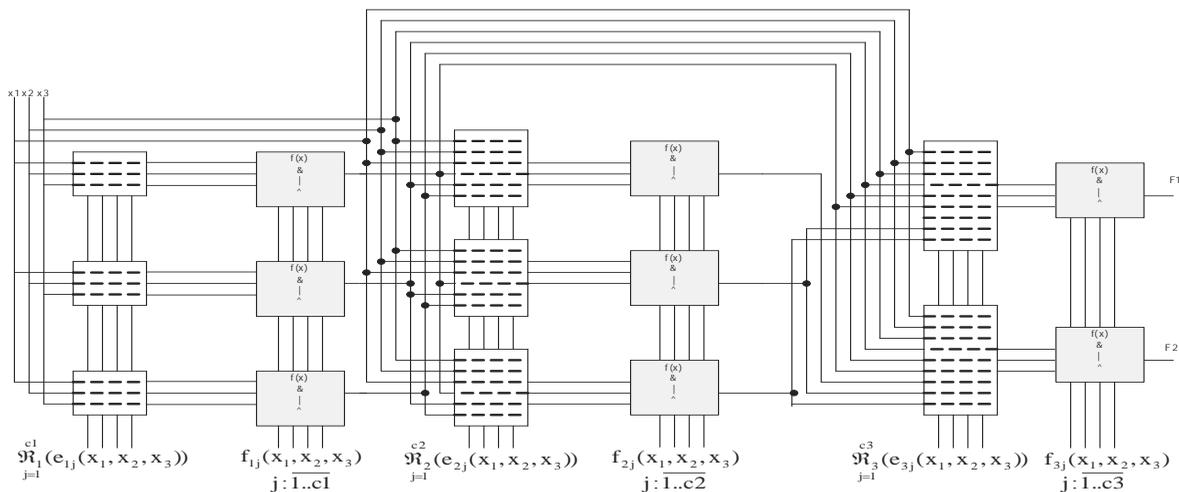
Figure 5. Application schema: Finding optimal solution for target function implementation
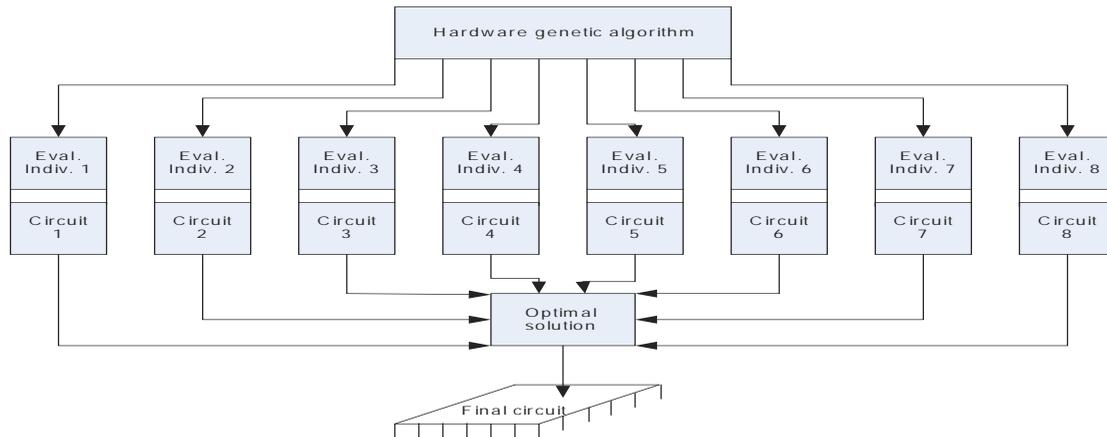
The last applications use dynamic target functions. Here each individual represent a complete solution for circuit. Evolution loop here is in two steps. First step is same like in the first application: loop until solution is found. After the solution is found in an individual named main individual the evolution continue for the others individuals. The target of the second step in evolution loop is to obtain different individuals relative to the main individual. When the target function is changed, the evolution loop pass in first step and the individuals, with high degree of dispersion, evolve to new solution.

CONCLUSIONS

In this paper we have presented the concept of the evolvable hardware and show a practical implementation of hardware genetic algorithm and reconfigurable hardware structure.

Hardware genetic algorithm increases the convergence speed to solutions which represent configuration for reconfigurable circuit.

It can be used for evolvable synthesis of digital circuit in intrinsic evolvable hardware. The bit string solutions which are giving by genetic algorithm can be connections configuration for a dynamic reconfigurable hardware circuit.

We present here three architectures of reconfigurable circuits which can be dynamically programmed by same hardware genetic algorithm module. The structure was implemented on Xilinx FPGA Spartan 3.

FUTURE TRENDS

There are more directions of research from this paper. First is design of reconfigurable circuit by using FPGA primitives. The new generation of FPGA (Virtex5) allows dynamically reconfiguration using primitives. In this case the generic gate is replaced by physical cells from FPGA. Another direction is implementation of hybrid neuro-genetic structure. A hardware implementation neural network can be used to store the best solutions from genetic algorithm. This configuration can be used to improve convergence of genetic algorithm. The evolved hardware can be used to design analog circuits. In this case, Boolean reconfigurable circuit can be replacing by analog reconfigurable circuit (like Field Programmable Transistors Area).

REFERENCES

Ali B., Almaini A. and Kalganova T., "Evolutionary Algorithms and Their Use in the Design of Sequential

Logic Circuits", Springer – Genetic Programming and evolvable machines, vol.5, p. 11-29, Kluwer Academic Publisher, 2004.

Bland I. M. and Megson G.M., "Systolic Array Library for Hardware Genetic Algorithms", Parallel, Emergent and Distributed Architectures Laboratory, Department of Computer Science, University of Reading, 2001.

Coello C. A., Van Veldhuizen D. A. and Lamont G. B., "Evolutionary Algorithms for Solving Multi-Objective Problems", Kluwer Academic Publishers, New York, 2002.

Goldberg D. E., Kargupta H., Horn J. and Cant´u-Paz E., "Critical deme size for serial and parallel genetic algorithms", IlliGAL, University of Illinois, Jan. 1995.

Iana G. V., Serban G., Angelescu P., Ionescu L. and Mazare A., "Aspects on sigma-delta modulators implementation in hardware structures", Advances in Intelligent Systems and Technologies, Proceedings ECIT2006. European Conference on Intelligent Systems and Technologies, Iasi 2006.

Ionescu L., Serban G., Ionescu V., Anghelescu P. and Iana G., "Implementation of GAs în Reconfigurable Gates Network", Third European Conference on Intelligent Systems and Technologies ECIT 2004, ISBN 973-7994-78-7, 2004.

Koza J. R., Bennett III F. H., Hutchings J. L., Bade S. L., Keane M. A. and D. Andre, "Evolving sotring networks using genetic programming and the rapidly reconfigurable xilinx 6216 field programmable gate array," in Proc. 31st Asilomar Conf. Signals, Systems, and Comp., IEEE Press: New York, 1997.

Martin P., "A hardware implementation of a genetic programming system using FPGAs and HandelC", Springer Genetic Programming and Evolvable Machines, vol. 2, nr.4, p.317-343, 2001

Miller J., Job D. and Vassiliev V., "Principles in the evolutionary design of digital circuits – Part 1,2", Springer – Genetic Programming and Evolvable machines, vol. 1, p. 7-35, p. 259 – 288, Kluwer Academic Publishers, 2000.

Scott D., Seth S. and A. Samal, "A hardware engine for genetic algorithms," Technical Report UNL-CSE-97-001, Dept. Computer Science and Engineering, University of Nebraska-Lincon, 4 July, 1997.

Shaaban N., Hasegawa S. and Suzuky A., "Improvement of energy characteristics of CdZnTe semiconductors detectors", Genetic Programming and Evolvable Machines, vol.2. nr.3 289-299, Kluwer Academic Publisher, 2001.

Sharabi S. and Sipper M., "GP-Sumo: Using genetic programming to evolve sumobots", Springer. Genetic Programming and Evolvable machines, vol. 7, p.211-230, Springer Science+Business Media, 2006 .

Thompson A. and Layzell P., "Analysis of unconventional evolved electronics," Commun. ACM, 42(4), pp. 71–79, 1999.

Yasunaga M., Kim J., Yoshihara I., "Evolvable reasoning hardware: its prototyping and performance evaluation", Springer – Genetic Programming and Evolvable machines, vol. 2, p. 211-230, Kluwer Academic Publishers, 2001.

Zhao S., Jiao L., "Multi-objective evolutionary design and knowledge discovery of logic circuits based on an adaptive genetic algorithm", Springer. Genetic Programming and Evolvable machines, vol.7, p.195-210, Springer Science+Business Media, 2006.

KEY TERMS

Evolvable Hardware: Reconfigurable circuit which is programmed by evolved algorithm like GA. To extrinsic evolvable hardware evolved algorithm run to host station outside of the reconfigurable circuit (PC). To intrinsic evolvable hardware evolved algorithm run inside the same system with reconfigurable circuit (even same chip).

Genetic Algorithms (GA): A genetic algorithm (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems. Genetic algorithms are categorized as a stochastic local search technique. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). Individuals, with coding schema, are initial random values. All individuals are, in the first

step of algorithm evaluated to get the fitness value. In the next step, they are sorted by fitness and selected for genetic operators. The parents are the individuals involved in genetic operators like crossover or mutation. The offspring resulted are evaluated together with parents and the algorithm resume with the first step. The loop is repeated until the solution is find or the number of generation reach the limit given by the programmer.

Genotype: Describe the genetic constitution of an individual, that is the specific allelic makeup of an individual. In evolvable hardware it consist in a vector of configuration bits.

HGA: Hardware genetic algorithm is a hardware implementation of genetic algorithm. Hardware implementation increases the performance of the algorithm by replacing serial software modules with parallel hardware.

Microstructure: Integration of structure in same chip. Evolvable hardware microstructure is an intrinsic evolvable hardware with all modules in same chip.

Phenotype: Describe one of the traits of an individual that is measurable and that is expressed in only a subset of the individuals within that population. In evolvable hardware phenotype consist in the circuit coded by an individual.

Reconfigurable Circuit: Hardware structure consist in logical cell network which allow configuration of the interconnections between cells

Evolving Graphs for ANN Development and Simplification

Daniel Rivero
University of A Coruña, Spain

David Periscal
University of A Coruña, Spain

INTRODUCTION

One of the most successful tools in the Artificial Intelligence (AI) world is Artificial Neural Networks (ANNs). This technique is a powerful tool used in many different environments, with many different purposes, like classification, clustering, signal modelization, or regression (Haykin, 1999). Although they are very easy to use, their creation is not a simple task, because the expert has to do much effort and spend much time on it.

The development of ANNs can be divided into two parts: architecture development and training and validation. The architecture development determines not only the number of neurons of the ANN, but also the type of the connections among those neurons. The training determines the connection weights for such architecture.

The architecture design task is usually performed by means of a manual process, meaning that the expert has to test different architectures to find the one able to achieve the best results. Each architecture trial means training and validating it, which can be a process that needs many computational resources, depending on the complexity of the problem. Therefore, the expert has much participation in the whole ANN development, although techniques for relatively automatic creation of ANNs have been recently developed.

BACKGROUND

ANN development is a research topic that has attracted many researchers from the world of evolutionary algorithms (Nolfi & Parisi D., 2002) (Cantú-Paz & Kamath, 2005). These techniques follow the general strategy of an evolutionary algorithm: an initial population with different types of genotypes encoding also different

parameters – commonly, the connection weights and/or the architecture of the network and/or the learning rules – is randomly created and repeatedly induced to evolve.

The most direct application of EC tools in the ANN world is to perform the evolution of the weights of the connections. This process starts from an ANN with an already determined topology. In this case, the problem to be solved is the training of the connection weights, attempting to minimise the network failure. Most of training algorithms, as backpropagation (BP) algorithm (Rumelhart, Hinton & Williams, 1986), are based on gradient minimisation, which presents several inconveniences. The main of these disadvantages is that, quite frequently, the algorithm gets stuck into a local minimum of the fitness function and it is unable to reach a global minimum. One of the options for overcoming this situation is the use of an evolutionary algorithm, so the training process is done by means of the evolution of the connection weights within the environment defined by both, the network architecture, and the task to be solved. In such cases, the weights can be represented either as the concatenation of binary values or of real numbers on a genetic algorithm (GA) (Greenwood, 1997).

The evolution of architectures consists on the generation of the topological structure, i.e., establishing the connectivity and the transfer function of each neuron. To achieve this goal with an evolutionary algorithm, it is needed to choose how to encode the genotype of a given network for it to be used by the genetic operators.

The most typical approach is called direct encoding. In this technique there is a one-to-one correspondence between each of the genes and a determined part of the network. A binary matrix represents an architecture where every element reveals the presence or absence

of connection between two nodes (Alba, Aldana & Troya, 1993).

In comparison with direct encoding, there are some indirect encoding methods. In these methods, only some characteristics of the architecture are encoded in the chromosome. These methods have several types of representation.

Firstly, the parametric representations represent the network as a group of parameters such as number of hidden layers, number of nodes for each layer, number of connections between two layers, etc (Harp, Samad & Guha, 1989). Another non direct representation type is based on a representation system that uses grammatical rules (Kitano, 1990), shaped as production rules that make a matrix that represents the network.

Another type of encoding is the growing methods. In this case, the genotype contains a group of instructions for building up the network (Nolfi & Parisi, 2002).

All of these methods evolve architectures, either alone (most commonly) or together with the weights. The transfer function for every node of the architecture is supposed to have been previously fixed by a human expert and is the same for all the nodes of the network or, at least, all the nodes of the same layer. Only few methods that also induce the evolution of the transfer function have been developed (Hwang, Choi & Park, 1997).

ANN DEVELOPMENT WITH GENETIC PROGRAMMING

This section very briefly shows an example of how to develop ANNs using an AI tool, Genetic Programming (GP), which performs an evolutionary algorithm, and how it can be applied to Data Mining tasks.

Genetic Programming

GP (Koza, 92) is based on the evolution of a given population. Its working is similar to a GA. In this population, every individual represents a solution for a problem that is intended to be solved. The evolution is achieved by means of the selection of the best individuals – although the worst ones have also a little chance of being selected – and their mutual combination for creating new solutions. After several generations, the population is expected to contain some good solutions for the problem.

The GP encoding for the solutions is tree-shaped, so the user must specify which are the terminals (leaves of the tree) and the functions (nodes capable of having descendants) for being used by the evolutionary algorithm in order to build complex expressions.

The wide application of GP to various environments and its consequent success are due to its capability for being adapted to numerous different problems. Although the main and more direct application is the generation of mathematical expressions (Rivero, Rabuñal, Dorado & Pazos, 2005), GP has been also used in other fields such as filter design (Rabuñal, Dorado, Puertas, Pazos, Santos & Rivero D., 2003), knowledge extraction, image processing (Rivero, Rabuñal, Dorado & Pazos, 2004), etc.

Model Overview

This work will use a graph-based codification to represent ANNs in the genotype. These graphs will not contain any cycles. Due to this type of codification the genetic operators had to be changed in order to be able to use the GP algorithm. The operators were changed in this way:

- The creation algorithm must allow the creation of graphs. This means that, at the moment of the creation of a node's child, this algorithm must allow not only the creation of this node, but also a link to an existing one in the same graph, without making cycles inside the graph.
- The crossover algorithm must allow the crossing of graphs. This algorithm works very similar to the existing one for trees, i.e. a node is chosen on each individual to change the whole subgraph it represents to the other individual. Special care has to be taken with graphs, because before the crossover there may be links from outside this subgraph to any nodes on it. In this case, after the crossover these links are updated and changed to point to random nodes in the new subgraph.
- The mutation algorithm has been changed too, and also works very similar to the GP tree-based mutation algorithm. A node is chosen from the individual and its subgraph is deleted and replaced with a new one. Before the mutation occurs, there may be nodes in the individual pointing to other nodes in the subgraph. These links are updated

Table 1. Summary of the operators to be used in the tree

Node	Type	Num. Children	Children type
ANN	ANN	Num. outputs	NEURON
n-Neuron	NEURON	2*n	n NEURON
			n REAL (weights)
n-Input	NEURON	0	-
+,-,*,%	REAL	2	REAL
[-4,4]	REAL	0	-

and made to point to random nodes in the new subgraph.

These algorithms must also follow two restrictions in GP: typing and maximum height. The GP typing property (Montana, 1995) means that each node will have a type and will also provide which type will have each of its children. This property provides the ability of developing structures that follow a specific grammar.

In order to be able to use GP to develop any kind of system, it is necessary to specify the set of operators that will be in the tree. With them, the evolutionary system must be able to build correct trees that represent ANNs. An overview of the operators used can be seen on Table 1.

This table shows a summary of the operators that can be used in the tree. This set of terminals and functions are used to build a tree that represents an ANN.

Figure 1. GP graph and its resulting network
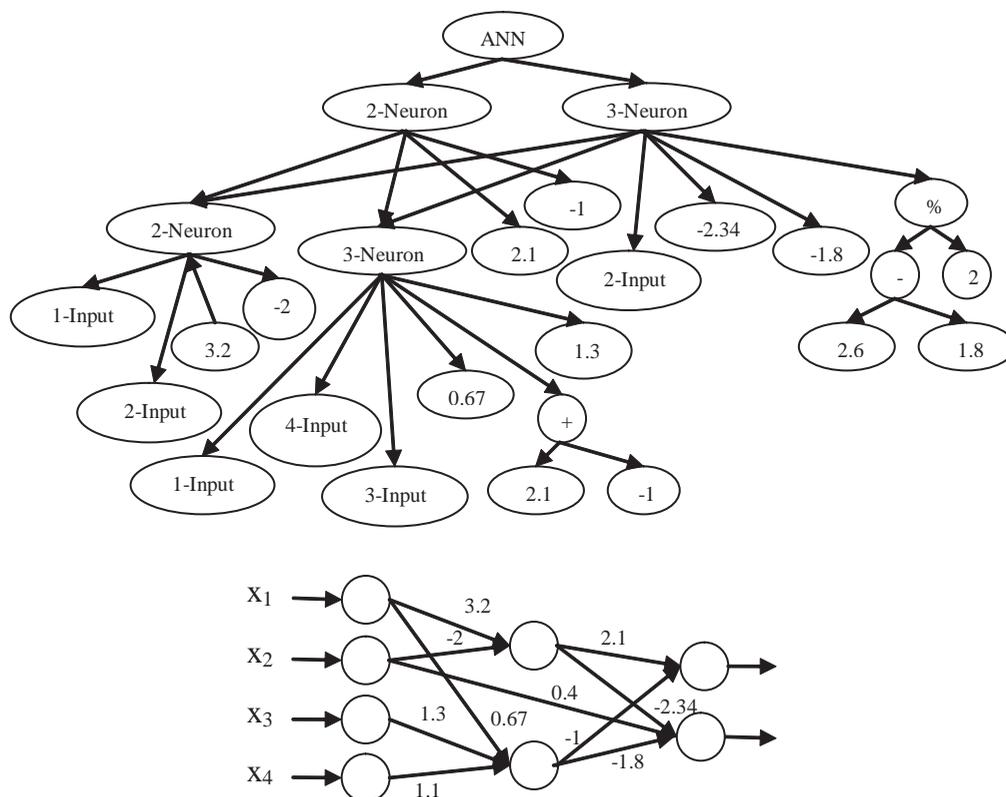
Although these sets are not explained in the text, in Fig. 1 can be seen an example of how they can be used to represent an ANN.

These operators are used to build GP trees. These trees have to be evaluated, and, once the tree has been evaluated, the genotype turns into phenotype. In other words, it is converted into an ANN with its weights already set (thus it does not need to be trained) and therefore can be evaluated. The evolutionary process demands the assignation of a fitness value to every genotype. Such value is the result of the evaluation of the network with the pattern set that represents the problem. This result is the Mean Square Error (MSE) of the difference between the network outputs and the desired outputs. Nevertheless, this value has been modified in order to induce the system to generate simple networks. The modification has been made by adding a penalization value multiplied by the number of neurons of the network. In such way, and given that the evolutionary system has been designed in order to minimise an error value, when adding a fitness value, a larger network would have a worse fitness value. Therefore, the existence of simple networks would be preferred as the penalization value that is added is proportional to the number of neurons at the ANN. The calculus of the final fitness will be as follows:

$$fitness = MSE + N * P$$

where N is the number of neurons of the network and P is the penalization value for such number.

Example of Applications

This technique has been used for solving problems of different complexity taken from the UCI (Mertz & Murphy, 2002). All these problems are knowledge-extraction problems from databases where, taking certain features as a basis, it is intended to perform a prediction about another attribute of the database. A small description of the problems to be solved can be seen at Table 2, along with other ANN parameters used later in this work.

All these databases have been normalised between 0 and 1 and divided into two parts, taking the 70% of the data base for training and using the remaining 30% for performing tests.

Results and Comparison with Other Methods

Several experiments have been performed in order to evaluate the system performance. The values taken for the parameters at these experiments were the following:

- Population size: 1000 individuals.
- Crossover rate: 95%.
- Mutation probability: 4%.
- Selection algorithm: 2-individual tournament.
- Graph maximum height: 5.
- Maximum inputs for each neuron: 9.
- Penalization value: 0.00001.

To achieve these values, several experiments had to be done in order to obtain values for these parameters that would return good results to all of the problems. These problems are very different in complexity, so it is expected that these parameters give good results to many different problems.

In order to evaluate its performance, the system presented here has been compared with other ANN generation and training methods.

The method 5x2cv was used by Cantú-Paz and Kamath (1995) for the comparison of different ANN generation and training techniques based on EC tools. This work presents as results the average precisions obtained in the 10 test results generated by this method. Such values are the basis for the comparison of the technique described here with other well known ones, described in detail by Cantú-Paz and Kamath (1995). Such work shows the average times needed to achieve the results. Not having the same processor that was used, the computational effort needed for achieving the results can be estimated. This effort represents the number of times that the pattern file was evaluated. The computational effort for every technique can be measured using the population size, the number of generations, the number of times that the BP algorithm was applied, etc. This calculation varies for every algorithm used. All the techniques that are compared with the work are related to the use of evolutionary algorithms for ANN design. Five iterations of a 5-fold crossed validation test were performed in all these techniques in order to evaluate the accuracy of the networks. These techniques are connectivity matrix, pruning, parameter search and graph-rewriting grammar.

Table 2 shows a summary of the number of neurons used by Cantú-Paz and Kamath (1995) in order to solve the problems that were used with connectivity matrix and pruning techniques. The epoch number of the BP algorithm, when used, is also indicated here.

Table 3 shows the parameter configuration used by these techniques. The execution was stopped after 5 generations with no improvement or after 50 total generations.

The results obtained with these 4 methods are shown in Table 4. Every box of the table indicates 3 different values: precision value obtained by Cantú-Paz and Kamath (1995) (left), computational effort needed for obtaining such value with that technique (below) and

Table 2. Summary of the problems to be solved

Problem	Description			ANN configuration			
	Number of inputs	Number of instances	Number of outputs	Inputs	Hidden	Outputs	BP Epochs
Breast Cancer	9	699	1	9	5	1	20
Iris Flower	4	150	3	4	5	3	80
Heart Disease	13	303	1	26	5	1	40
Ionosphere	34	351	1	34	10	1	40

Table 3. Parameters of the techniques used for the comparison

Parameter	Matrix	Pruning	Parameters	Grammar
Chromosome length (L)	N	N	36	256
Population size	$\lfloor 3\sqrt{L} \rfloor$	$\lfloor 3\sqrt{L} \rfloor$	25	64
Crossover points	L/10	L/10	2	L/10
Mutation rate	1/L	1/L	0.04	0.004

N = (hidden+output)*input + output*hidden

Table 4. Comparison with other methods

Problem	Matrix		Pruning		Parameters		Grammar	
Breast Cancer	96.77	96.27	96.31	95.79	96.69	96.27	96.71	96.31
		92000		4620		100000		300000
Iris Flower	92.40	95.49	92.40	81.58	91.73	95.52	92.93	95.66
		320000		4080		400000		1200000
Heart Cleveland	76.78	81.11	89.50	78.28	65.89	81.05	72.8	80.97
		304000		7640		200000		600000
Ionosphere	87.06	88.34	83.66	82.37	85.58	87.81	88.03	88.36
		464000		11640		200000		600000
Average	88.25	90.30	90.46	84.50	84.97	90.16	87.61	90.32

precision value obtained with the technique described here and related to the previously mentioned computational effort value (right).

Watching this table, it is obvious that the results obtained with the method proposed here are, not only similar to the ones presented by Cantú-Paz and Kamath (1995), but better in most of the cases. The reason of this lies in the fact that these methods need a high computational load since training is necessary for every case of network (individual) evaluation, which therefore turns to be time-consuming. During the work described here, the procedures for design and training are performed simultaneously, and therefore, the times needed for designing as well as for evaluating the network are combined.

FUTURE TRENDS

The future line of works in this area would be the study of the system parameters in order to evaluate their impact on the results from different problems.

Another interesting line consists on the combination of this graph evolution algorithm with a GA that performs an optimization process on the weight values. With this modification, the whole system will have two levels:

1. The graph evolution algorithm explained in this work performs the evolution of the architectures.
2. The GA takes those architectures and optimizes the weights of the connections.

With this architecture, the evolution of ANNs can be seen as a lamarckian strategy.

CONCLUSION

This work describes a technique in which an evolutionary algorithm is used to automatically develop ANNs. This evolutionary algorithm performs graph evolution, and it is based on the GP algorithm, although it had to be modified in order to make it operate with graphs instead of trees.

Results show that the networks returned by this algorithm give, in most of the cases, an error lower than the error given by the rest of the ANN development systems used for the comparison. Only one technique (pruning) performs better that the one described here. However, that technique still needs some work from the expert, to do the design of the initial network.

Most of the techniques used for the ANN development are quite costly, due in some cases to the combination of training with architecture evolution. The technique described here is able to achieve good results with a low computational cost and besides, the added advantage is that, not only the architecture and the connectivity of the network are evolved, but also the network itself undergoes an optimization process.

ACKNOWLEDGMENT

REFERENCES

Alba E., Aldana J.F. & Troya J.M. (1993) Fully automatic ANN design: A genetic approach. *Proc. Int. Workshop Artificial Neural Networks (IWANN'93), Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 686, 399-404.

Cantú-Paz E. & Kamath C. (2005) An Empirical Comparison of Combinatios of Evolutionary Algorithms and Neural Networks for Classification Problems. *IEEE Transactions on systems, Man and Cybernetics – Part B: Cybernetics*. 915-927.

Greenwood G.W. (1997) Training partially recurrent neural networks using evolutionary strategies. *IEEE Trans. Speech Audio Processing*, 5, 192-194.

Harp S.A., Samad T. & Guha A. (1989) Toward the genetic synthesis of neural networks. *Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications*, J.D. Schafer, Ed. San Mateo, CA: Morgan Kaufmann. 360-369.

Haykin, S. (1999). *Neural Networks (2nd ed.)*. Englewood Cliffs, NJ: Prentice Hall.

Hwang M.W., Choi J.Y. & Park J. (1997) Evolutionary projection neural networks. *Proc. 1997 IEEE Int. Conf. Evolutionary Computation, ICEC'97*. 667-671.

Jung-Hwan Kim, Sung-Soon Choi & Byung-Ro Moon (2005) Normalization for neural network in genetic search. *Genetic and Evolutionary Computation Conference*. 1-10.

Kitano H. (1990) Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4, 461-476.

Koza, J. R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Mertz C.J. & Murphy P.M. (2002). UCI repository of machine learning databases. http://www-old.ics.uci.edu/pub/machine-learning-databases.

Montana D.J. (1995) Strongly typed genetic programming. *Evolutionary Computation*, 3(2), 199-200.

Nolfi S. & Parisi D. (2002) Evolution of Artificial Neural Networks. *Handbook of brain theory and neural networks, Second Edition*. Cambridge, MA: MIT Press. 418-421.

Rabuñal J.R., Dorado J., Puertas J., Pazos A., Santos A. & Rivero D. (2003) Prediction and Modelling of the Rainfall-Runoff Transformation of a Typical Urban Basin using ANN and GP. *Applied Artificial Intelligence*.

Rivero D., Rabuñal J.R., Dorado J. & Pazos A. (2004) Using Genetic Programming for Character Discrimination in Damaged Documents. *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*. 349-358.

Rivero D., Rabuñal J.R., Dorado J. & Pazos A. (2005) Time Series Forecast with Anticipation using Genetic Programming. *IWANN 2005*. 968-975.

Rumelhart D.E., Hinton G.E. & Williams R.J. (1986) Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*. D. E. Rumelhart & J.L. McClelland, Eds. Cambridge, MA: MIT Press. 1, 318-362.

KEY TERMS

Artificial Neural Networks: Interconnected set of many simple processing units, commonly called neurons, that use a mathematical model, that represents an input/output relation,

Back-Propagation Algorithm: Supervised learning technique used by ANNs, that iteratively modifies the weights of the connections of the network so the error given by the network after the comparison of the outputs with the desired one decreases.

Evolutionary Computation: Set of Artificial Intelligence techniques used in optimization problems, which are inspired in biologic mechanisms such as natural evolution.

Genetic Programming: Machine learning technique that uses an evolutionary algorithm in order to optimise the population of computer programs according to a fitness function which determines the capability of a program for performing a given task.

Genotype: The representation of an individual on an entire collection of genes which the crossover and mutation operators are applied to.

Phenotype: Expression of the properties coded by the individual's genotype.

Population: Pool of individuals exhibiting equal or similar genome structures, which allows the application of genetic operators.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in.

Facial Expression Recognition for HCI Applications

Fadi Dornaika
Institut Géographique National, France

Bogdan Raducanu
Computer Vision Center, Spain

INTRODUCTION

Facial expression plays an important role in cognition of human emotions (Fasel, 2003 & Yeasin, 2006). The recognition of facial expressions in **image sequences** with significant head movement is a challenging problem. It is required by many applications such as human-computer interaction and computer graphics animation (Cañamero, 2005 & Picard, 2001). To classify expressions in still images many techniques have been proposed such as Neural Nets (Tian, 2001), Gabor wavelets (Bartlett, 2004), and active appearance models (Sung, 2006). Recently, more attention has been given to modeling facial deformation in dynamic scenarios. Still image classifiers use feature vectors related to a single frame to perform classification. **Temporal classifiers** try to capture the temporal pattern in the sequence of feature vectors related to each frame such as the Hidden Markov Model based methods (Cohen, 2003, Black, 1997 & Rabiner, 1989) and Dynamic Bayesian Networks (Zhang, 2005). The main contributions of the paper are as follows. First, we propose an efficient recognition scheme based on the detection of **keyframes** in videos where the recognition is performed using a temporal classifier. Second, we use the proposed method for extending the human-machine interaction functionality of a robot whose response is generated according to the user's recognized facial expression.

Our proposed approach has several advantages. First, unlike most expression recognition systems that require a frontal view of the face, our system is view- and texture-independent. Second, its learning phase is simple compared to other techniques (e.g., the Hidden Markov Models and Active Appearance Models), that is, we only need to fit second-order Auto-Regressive models to sequences of facial actions. As a result, even when the imaging conditions change the learned Auto-Regressive models need not to be recomputed.

The rest of the paper is organized as follows. Section 2 summarizes our developed appearance-based 3D **face tracker** that we use to track the 3D **head pose** as well as the **facial actions**. Section 3 describes the proposed facial expression recognition based on the detection of keyframes. Section 4 provides some experimental results. Section 5 describes the proposed human-machine interaction application that is based on the developed facial expression recognition scheme.

SIMULTANEOUS HEAD AND FACIAL ACTION TRACKING

In our study, we use the *Candide* **3D face model** (Ahlberg, 2001). This 3D deformable wireframe model is given by the 3D coordinates of n vertices. Thus, the 3D shape can be fully described by the $3n$-vector $\boldsymbol{g}$ - the concatenation of the 3D coordinates of all vertices. The vector $\boldsymbol{g}$ can be written as:

$$g = g_s + A\tau_a \qquad (1)$$

where $\boldsymbol{g}_s$ is the static shape of the model, τ_a is the facial action vector, and the columns of A are the Animation Units. In this study, we use six modes for the facial Animation Units (AUs) matrix A, that is, the dimension of τ_a is 6. These modes are all included in the *Candide* model package. We have chosen the six following AUs: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lowerer and outer eyebrow raiser. A cornerstone problem in **facial expression** recognition is the ability to track the local **facial actions**/deformations. In our work, we track the head and facial actions using our **face tracker** (Dornaika & Davoine, 2006). This appearance-based tracker simultaneously computes the 3D **head pose** and the **facial actions** τ_a by minimizing a distance between

the incoming warped frame and the current appearance of the face. Since the **facial actions,** encoded by the vector τ_a, are highly correlated to the facial expressions, their time series representation can be utilized for inferring the facial expression in videos. This will be explained in the sequel.

EFFICIENT FACIAL EXPRESSION DETECTION AND RECOGNITION

In (Dornaika & Raducanu, 2006), we have proposed a **facial expression** recognition method that is based on the time-series representation of the tracked facial actions τ_a. An analysis-synthesis scheme based on learned auto-regressive models was proposed. In this paper, we introduce a process able to detect **keyframes**

in videos. Once a keyframe is detected, the temporal recognition scheme described in (Dornaika & Raducanu, 2006) will be invoked on the detected keyframe. The proposed scheme has two advantages. First, the CPU time corresponding to the recognition part will be considerably reduced since only few keyframes are considered. Second, since a **keyframe** and its neighbor frames are characterizing the expression, the discrimination performance of the recognition scheme will be boosted. In our case, the keyframes are defined by the frames where the **facial actions** change abruptly. Thus, a keyframe can be detected by looking for a local positive maximum in the temporal derivatives of the facial actions. To this end, two entities will be computed from the sequence of facial actions τ_a that arrive in a sequential fashion: (i) the L_1 norm $\|\tau_a\|_1$, and (ii) the temporal derivative given by:

Figure 1. Efficient facial expression detection and recognition based on keyframes

Figure 2. Keyframe detection and recognition applied on a 1600-frame sequence
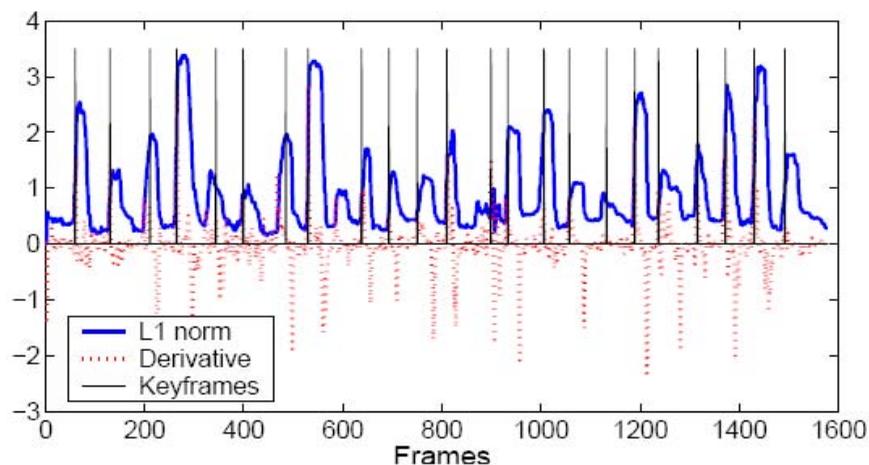
$$D_t = \frac{\partial \|\tau_a\|_1}{\partial t} = \sum_{i=1}^{6} \frac{\partial \tau_{a(i)}}{\partial t} \qquad (2)$$

In the above equation, we have used the fact that the facial actions are positive. Let W be the size of a temporal segment defining the temporal granulometry of the system. In other words, the system will detect and recognize at most one expression every W frames. In practice, W belongs to [0.5s, 1s]. The whole scheme is depicted in Figure 1.

In this figure, we can see that the system has three levels: the tracking level, the keyframe detection level, and the recognition level. The tracker provides the facial actions for every frame. Whenever the current video segment size reaches W frames, the keyframe detection is invoked to select a keyframe in the current segment if any. A given frame is considered as a **keyframe** if it meets three conditions: (1) the corresponding D_t is a positive local maximum (within the segment), (2) the corresponding norm $\|\tau_a\|_1$ is greater than a predefined threshold, (3) its far from the previous keyframe by at least W frames. Once a keyframe is found in the current segment, the dynamical classifier described in (Dornaika & Raducanu, 2006) will be invoked.

Figure 2 shows the results of applying the proposed detection scheme on a 1600-frame sequence containing 23 played expressions. Some images are shown in Figure 4. The solid curve corresponds to the norm $\|\tau_a\|_1$, the dotted curve to the derivative D_t and the vertical bars correspond to the detected keyframes. In this example, the value of W is set to 30 frames. As can be seen, out of 1600 frames only 23 keyframes will be processed by the expression classifier.

EXPERIMENTAL RESULTS

Recognition results: We used a 300-frame video sequence. For this sequence, we asked a subject to display several expressions arbitrarily (see Figure 3). The middle of this figure shows the normalized similarities associated with each universal expression where the recognition is performed for every frame in the sequence. As can be seen, the **temporal classifier** (Dornaika & Raducanu, 2006) has correctly detected the presence of the surprise, joy, and sadness expressions. Note that the mixture of expressions at transition is normal since the recognition is performed in a frame-wise manner. The lower part of this figure shows the results of applying the proposed keyframe detection scheme. On a 3.2 GHz PC, a non-optimized C code of the developed approach carries out the tracking and recognition in about 60 ms.

Performance study: In order to quantify the recognition rate, we have used 35 test videos retrieved from the CMU database. Table 1 shows the confusion matrix associated with the 35 test videos featuring 7 persons. As can be seen, although the recognition rate was good (80%), it is not equal to 100%. This can be explained by the fact that the expression dynamics are highly subject-dependent. Recall that the used auto-regressive models are built using data associated with one subject. Notice that the human 'ceiling' in correctly classifying facial expressions into the six basic emotions has been established at 91.7%.

F

Table 1. Confusion matrix for the facial expression classifier associated with 35 test videos (CMU data). The model is built using one unseen person

	Surprise (7)	Sadness (7)	Joy (7)	Disgust (7)	Anger (7)
Surprise	7	0	0	0	0
Sadness	0	7	0	5	0
Joy	0	0	7	0	0
Disgust	0	0	0	2	2
Anger	0	0	0	0	5

Figure 3. Top: Four frames (50, 110, 150, and 250) associated with a 300-frame test sequence. Middle: The similarity measure computed for each universal expression and for each non-neutral frame of the sequence-the framewise recognition. Bottom: The recognition based on keyframe detection.
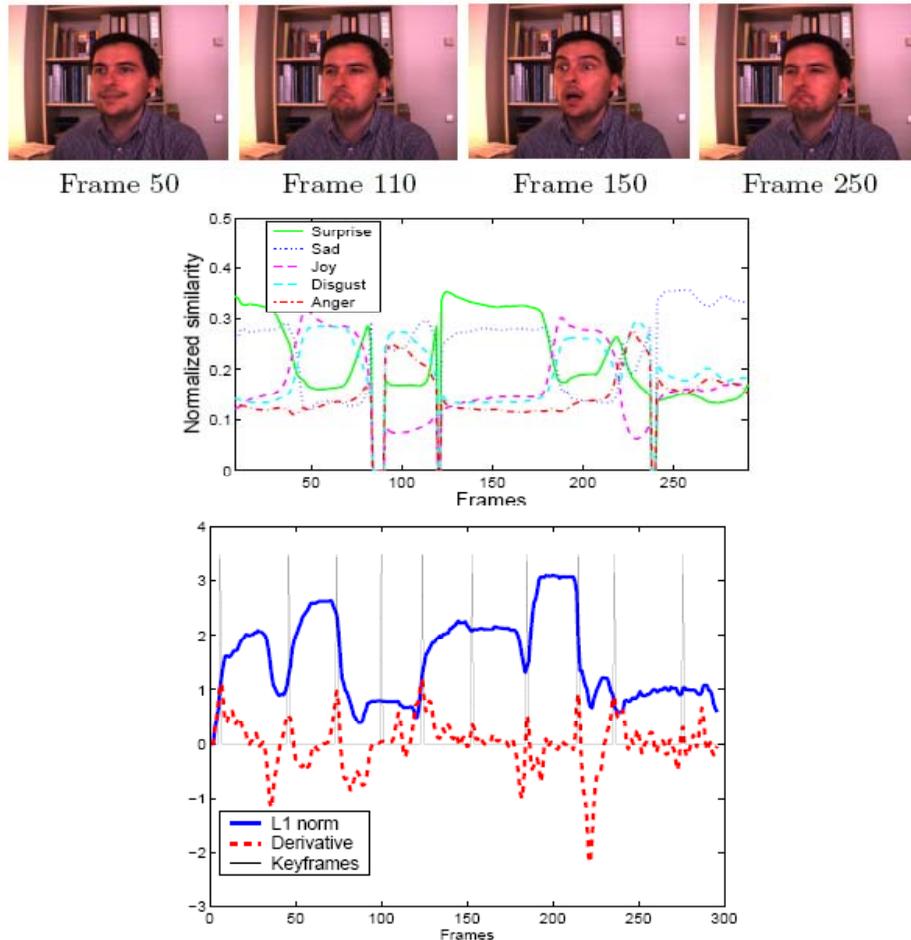
HUMAN-MACHINE INTERACTION

Interpreting non-verbal face gestures is used in a wide range of applications. An intelligent user-interface not only should interpret the face movements but also should interpret the user's emotional state (Breazeal, 2002). Knowing the emotional state of the user makes machines communicate and interact with humans in a natural way: intelligent entertaining systems for kids, interactive computers, intelligent sensors, social robots,

to mention a few. In the sequel, we will show how our proposed technique lends itself nicely to such applications. Without loss of generality, we use the AIBO robot which has the advantage of being especially designed for Human Computer Interaction. The input to the system is a video stream capturing the user's face.

The AIBO robot: AIBO is a biologically-inspired robot and is able to show its emotions through an array of LEDs situated in the frontal part of the head. In addition to the LEDs' configuration, the robot response

F

Figure 4. Top: Some detected keyframes associated with the 1600-frame video. Middle: The recognized expression. Bottom: The corresponding robot's response.
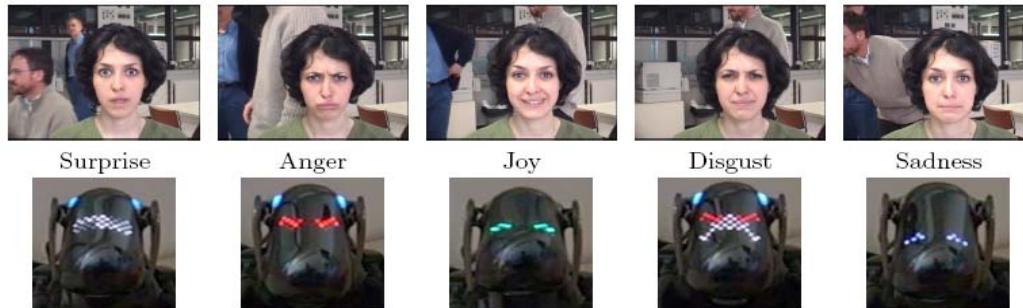
contains some small head and body movements. From its concept design, AIBO's affective states are triggered by the Emotion Generator engine. This occurs as a response to its internal state representation, captured through multi-modal interaction (vision, audio and touch). For instance, it can display the 'happiness' feeling when it detects a face (through the vision system) or it hears a voice. But it does not possess a built-in system for vision-based automatic facial-expression recognition. For this reason, with the scheme proposed in this paper (see Section 3), we created an application for AIBO whose purpose is to enable it with this capability.

This application is a very simple one, in which the robot is just imitating the expression of a human subject. Usually, the response of the robot occurs slightly after the apex of the human expression. The results of this application were recorded in a 2 minute video which can be downloaded from the following address: http://www.cvc.uab.es/~bogdan/AIBO-emotions.avi. In order to be able to display simultaneously in the video the correspondence between subject's and robot's expressions, we put them side by side.

Figure 4 illustrates five detected keyframes from the 1600 frame video depicted in Figure 2. These are shown in correspondence with the robot's response. The middle row shows the recognized expression. The bottom row shows a snapshot of the robot head when it interacts with the detected and recognized expression.

CONCLUSION

This paper described a view- and texture-independent approach to facial expression analysis and recognition. The paper presented two contributions. First, we proposed an efficient facial expression recognition scheme based on the detection of keyframes in videos. Second, we applied the proposed method in a Human Computer Interaction scenario, in which an AIBO robot is mirroring the user's recognized facial expression.

ACKNOWLEDGMENT

REFERENCES

Ahlberg, J. (2001). CANDIDE-3 – An Updated Parameterized Face. *Technical Report LiTH-ISY-R-2326*, Dept. of Electrical Engineering, Linköping University, Sweden.

Bartlett, M., Littleworth, G., Lainscsek, C., Fasel I. & Movellan, J. (2004). Machine Learning Methods for Fully Automatic Recognition of Facial Expressions and Facial Actions. *Proc. of IEEE Conference on Systems, Man and Cybernetics*, Vol. I, The Hague, The Netherlands, pp.592-597.

Black, M.J. & Yacoob, Y. (1997). Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23-48.

Breazeal, C. & Scassellati, B. (2002). Robots that Imitate Humans. *Trends in Cognitive Science*, Vol. 6, pp. 481-487.

Cañamero, L. & Gaussier, P. (2005). Emotion Understanding: Robots as Tools and Models. In *Emotional Development: Recent Research Advances*, pp. 235-258.

Cohen, I., Sebe, N., Garg, A., Chen, L. & Huang, T. (2003). Facial Expression Recognition from Video Sequences: Temporal and Static Modeling. *Computer Vision and Image Understanding*, 91(1-2):160-187.

Dornaika, F. & Davoine, F. (2006). On Appearance Based Face and Facial Action Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(9):1107-1124.

Dornaika, F. & Raducanu, B. (2006). Recognizing Facial Expressions in Videos Using a Facial Action Analysis-Synthesis Scheme. Proc. of IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. N/A, Australia.

Fasel, B. & Luettin, J. (2003). Automatic Facial Expression Analysis: A Survey. *Pattern Recognition*, 36(1):259-275.

Picard, R., Vyzas, E. & Healy, J. (2001) Toward Machine Emotional Intelligence: Analysis of Affective Psychological State. *IEEE Trasactions on Pattern Analysis and Machine Intelligence*, 23(10):1175-1191.

Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of IEEE*, 77(2):257-286.

Sung, J., Lee, S. & Kim, D. (2006). A Real-Time Facial Expression Recognition Using the STAAM. *Proc. of International Conference on Pattern Recognition*, Vol. I, pp. 275-278, Hong-Kong.

Tian, Y., Kanade T. & Cohn, J. (2001). Recognizing Action Units for Facial Expression Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, pp. 97-115.

Yeasin M., Bullot, B. & Sharma, R. (2006). Recognition of Facial Expressions and Measurement of Levels of Interest from Video. *IEEE Transactions on Multimedia* 8(3):500-508.

Zhang, Y. & Ji, Q. (2005). Active and Dynamic Information Fusion for Facial Expression Understanding from Image Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):699-714.

KEY TERMS

3D Deformable Model: A model which is able to modify its shape while being acted upon by an external influence. In consequence, the relative position of any point on a deformable body can change.

Active Appearance Models (AAM): Computer Vision algorithm for matching a statistical model of object shape and appearance to a new image. The approach is widely used for matching and tracking faces.

AIBO: One of several types of robotic pets designed and manufactured by Sony. Able to walk, "see" its environment via camera, and recognize spoken commands, they are considered to be autonomous robots, since they are able to learn and mature based on external stimuli from their owner or environment, or from other AIBOs.

Autoregressive Models: Group of linear prediction formulas that attempt to predict the output of a system based on the previous outputs and inputs.

Facial Expression Recognition System: Computer-driven application for automatically identifying person's facial expression from a digital still or video image. It does that by comparing selected facial features in the live image and a facial database.

Hidden Markov Model (HMM): Statistical model in which the system being modeled is assumed to be

a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications.

Human–Computer Interaction (HCI): The study of interaction between people (users) and computers. It is an interdisciplinary subject, relating computer science with many other fields of study and research (Artificial Intelligence, Psychology, Computer Graphics, Design).

Social Robot: An autonomous robot that interacts and communicates with humans by following the social rules attached to its role. This definition implies that a social robot has a physical embodiment. A consequence of the previous statements is that a robot that only interacts and communicates with other robots would not be considered to be a social robot.

Wireframe Model: The representation of all surfaces of a three-dimensional object in outline form.

Feature Selection

Noelia Sánchez-Maroño
University of A Coruña, Spain

Amparo Alonso-Betanzos
University of A Coruña, Spain

INTRODUCTION

Many scientific disciplines use modelling and simulation processes and techniques in order to implement non-linear mapping between the input and the output variables for a given system under study. Any variable that helps to solve the problem may be considered as input. Ideally, any classifier or regressor should be able to detect important features and discard irrelevant features, and consequently, a pre-processing step to reduce dimensionality should not be necessary. Nonetheless, in many cases, reducing the dimensionality of a problem has certain advantages (Alpaydin, 2004; Guyon & Elisseeff, 2003), as follows:

- Performance improvement. The complexity of most learning algorithms depends on the number of samples and features (curse of dimensionality). By reducing the number of features, dimensionality is also decreased, and this may save on computational resources—such as memory and time—and shorten training and testing times.
- Data compression. There is no need to retrieve and store a feature that is not required.
- Data comprehension. Dimensionality reduction facilitates the comprehension and visualisation of data.
- Simplicity. Simpler models tend to be more robust when small datasets are used.

There are two main methods for reducing dimensionality: feature extraction and feature selection. In this chapter we propose a review of different feature selection (FS) algorithms, including its main approaches: filter, wrapper and hybrid – a filter/wrapper combination.

BACKGROUND

Feature extraction and feature selection are the main methods for reducing dimensionality. In feature extraction, the aim is to find a new set of r dimensions that are a combination of the n original ones. The best known and most widely used unsupervised feature extraction method is principal component analysis *(PCA)*; commonly used as supervised methods are linear discriminant analysis *(LDA)* and partial least squares *(PLS)*.

In feature selection, a subset of r relevant features is selected from a set n, whose remaining features will be ignored. As for the evaluation function used, FS approaches can be mainly classified as filter or wrapper models (Kohavi & John, 1997). Filter models rely on the general characteristics of the training data to select features, whereas wrapper models require a predetermined learning algorithm to identify the features to be selected. Wrapper models tend to give better results, but when the number of features is large, filter models are usually chosen because of their computational efficiency. In order to combine the advantages of both models, hybrid algorithms have recently been proposed (Guyon et al., 2006).

FEATURE SELECTION

The advantages described in the Introduction section denote the importance of dimensionality reduction. Feature selection is also useful when the following assumptions are made:

- There are inputs that are not required to obtain the output.
- There is a high correlation between some of the input features.

A feature selection algorithm (FSA) looks for an optimal set of features, and consequently, a paradigm that describes the FSA is heuristic search. Since each state of the search space is a subset of features, FSA can be characterised in terms of the following four properties (Blum & Langley, 1997):

- The initial state. This can be the empty set of features, the whole set or any random state.
- The search strategy. Although an exhaustive search leads to an optimal set of features, the associated computational and time costs are high when the number of features is high. Consequently, different search strategies are used so as to identify a good set of features within a reasonable time.
- The evaluation function used to determine the quality of each set of features. The goodness of a feature subset is dependent on measures. According to the literature, the following measures have been employed: information measures, distance measures, dependence measures, consistency measures, and accuracy measures.
- The stop criterion. An end point needs to be established; for example, the process should finish if the evaluation function has not improved after a new feature has been added/removed.

In terms of search method complexity, there are three main sub-groups (Salapa et al., 2007):

- Exponential strategies involving an exhaustive search of all feasible solutions. Exhaustive search guarantees identification of an optimal feature subset but has a high computational cost. Examples are the branch and bound algorithms.
- Sequential strategies based on a local search for solutions defined by the current solution state. Sequential search does not guarantee an optimal result, since the optimal solution could be in a region of the search space that is not searched. However, compared with exponential searching, sequential strategies have a considerably reduced computational cost. The best known strategies are sequential forward selection and sequential backward selection (SFS and SBS, respectively). SFS starts with an empty set of features and adds features one by one, while SBS begins with a full set and removes features one by one. Features are added or removed on the basis of improvements in the evaluation function. These approaches do not consider interactions between features, i.e., a feature may not reduce error by itself, but improvement may be achieved by the feature's link to another feature. Floating search (Pudil et al., 1994) solves this problem partially, in that the number of features included and/or removed at each stage is not fixed. Another approach (Sánchez et al., 2006) uses sensitivity indices (the importance of each feature is given in terms of the variance) to guide a backward elimination process, with several features discarded in one step.
- Random algorithms that employ randomness to avoid local optimal solutions and enable temporary transition to other states with poorer solutions. Examples are simulated annealing and genetic algorithms.

The most popular FSA classification, which refers to the evaluation function, considers the three (Blum & Langley, 1997) or last two (Kohavi & John, 1997) groups, as follows:

- Embedded methods. The induction algorithm is simultaneously an FSA. Examples of this method are decision trees, such as classification and regression trees (CART), and artificial neural networks (ANN).
- Filter methods. Selection is carried out as a pre-processing step with no induction algorithm (Figure 1). The general characteristics of the training data are used to select features (for example, distances between classes or statistical dependencies). This model is faster than the wrapper approach (described below) and results in a better generalisation because it acts independently of the induction algorithm. However, it tends to select subsets with a high number of features (even all the features) and so a threshold is required to choose a subset.
- Wrapper methods. Wrapper models use the induction algorithm to evaluate each subset of features, i.e., the induction algorithm is part of the evaluation function in the wrapper model, which means this model is more precise than the filter model. It also takes account of techniques, such as cross-validation, that avoid over-fitting. However, wrapper models are very time consuming, which

F

Figure 1. Filter algorithm
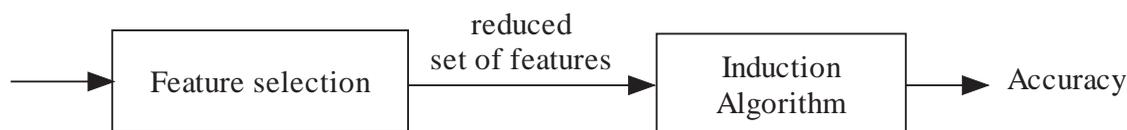
restricts application with some datasets. Moreover, although they may obtain good results with the inherent induction algorithm, they may perform poorly with an alternative algorithm.

Hybrid methods that combine filter and wrapper methods have recently been attracting a great deal of attention in the FS literature (Liu & Motoda, 1998; Guyon et al., 2006). Although the following sections of this chapter are mainly devoted to filter and wrapper methods, a brief review of the most recent hybrid methods is also included.

Filter Methods

A number of representative filter algorithms are described in the literature, such as χ^2-Statistic, information gain, or correlation based feature selection (CFS). For the sake of completeness, we will refer to two classical algorithms (FOCUS and RELIEF) and will describe very recently developed filter methods (FCBF and INTERACT). An exhaustive discussion of filter methods is provided in Guyon et al. (2006)—including of methods such as Random Forests (RF), an ensemble of tree classifiers.

FOCUS

In FOCUS (Almuallim & Dietterich, 1991) all feature subsets of increasing size are evaluated until a suitable subset is encountered. Feature subset q is said to be suitable if there is no pair of examples that have different class values and the same values for all the features in q. The successor of this algorithm is FOCUS_2 (Almuallim & Dietterich, 1992), which prunes the search space, thereby evaluating only promising subsets. FOCUS_2 is therefore much faster than FO-

CUS. However, using both algorithms in domains with a large number of features may be computationally unfeasible. Consequently, search heuristics are used in different versions of the algorithm, resulting in good but not necessarily optimal solutions.

RELIEF

The RELIEF algorithm (Kira & Rendell, 1992) estimates the quality of attributes according to how well their values distinguish between instances that are near to each other. For this purpose, given a randomly selected instance, $\mathbf{x_s} = \{x_{1s}, x_{2s}, \ldots, x_{ns}\}$, RELIEF searches for its two nearest neighbours: one from the same class, called *nearest hit H*, and the other from a different class, called *nearest miss M*. It then updates the quality estimate for all the features, depending on the values for $\mathbf{x_s}$, M, and H. RELIEF can deal with discrete and continuous features but is limited to two-class problems. An extension—ReliefF—not only deals with multiclass problems but is also more robust and capable of dealing with incomplete and noisy data. ReliefF was subsequently adapted for continuous class (regression) problems, resulting in the RReliefF algorithm (Robnik-Sikonja & Kononenko, 2003).

FCBF and INTERACT

The fast correlated-based filter (FCBF) method (Yu & Liu, 2003) is based on symmetrical uncertainty (SU), which is defined as the ratio between the information gain and the entropy of two features, x and y:

$$SU(x, y) = 2 \frac{IG(x / y)}{H(x) + H(y)}.$$

This method was designed for high-dimensionality data and has been shown to be effective in removing both irrelevant and redundant features. However, it fails to take into consideration the interaction between features. The INTERACT algorithm (Zhao & Liu, 2007) uses the same goodness measure, SU, but also includes the consistency contribution (c-contribution). It can thus handle feature interaction, and efficiently selects relevant features.

Wrapper Methods

The idea of the wrapper approach is to select a feature subset using a learning algorithm as part of the evaluation function (Figure 2). Instead of using subset sufficiency, entropy or another explicitly defined evaluation function, a kind of "black box" function is used to guide the search. The evaluation function for each candidate feature subset returns an estimate of the quality of the model that is induced by the learning algorithm. This can be rather time consuming, since, for each candidate feature subset evaluated during the search, the target learning algorithm is usually applied several times (e.g., in the case of 10-fold cross validation being used to estimate model quality). Here

we briefly describe several feature subset selection algorithms—developed in machine learning—that are based on the wrapper approach. The literature is vast in this area and so we will just focus on the most representative wrapper models.

An interesting study of the wrapper approach was conducted by Kohavi & John (1997). Besides introducing the notion of strong and weak feature relevance, these authors showed the results achieved by different induction algorithms (ID3, C4.5, and naïve Bayes) in several search methods (best first, hill-climbing, etc.). Aha & Bankert (1995) used a wrapper approach in instance-based learning and proposed a new search strategy that performs beam search using a kind of backward elimination; that is, instead of starting with an empty feature subset, the search randomly selects a fixed number of feature subsets and starts with the best among them. Caruana & Freitag (1994) developed a wrapper feature subset selection method for decision tree induction, proposing bidirectional hill-climbing for the feature space—as more effective than either forward or backward selection. Genetic algorithms have been broadly adopted to perform the search for the best subset of features in a wrapper way (Liu & Motoda, 1998, Huang et al. 2007). The feature selection

Figure 2. Wrapper algorithm
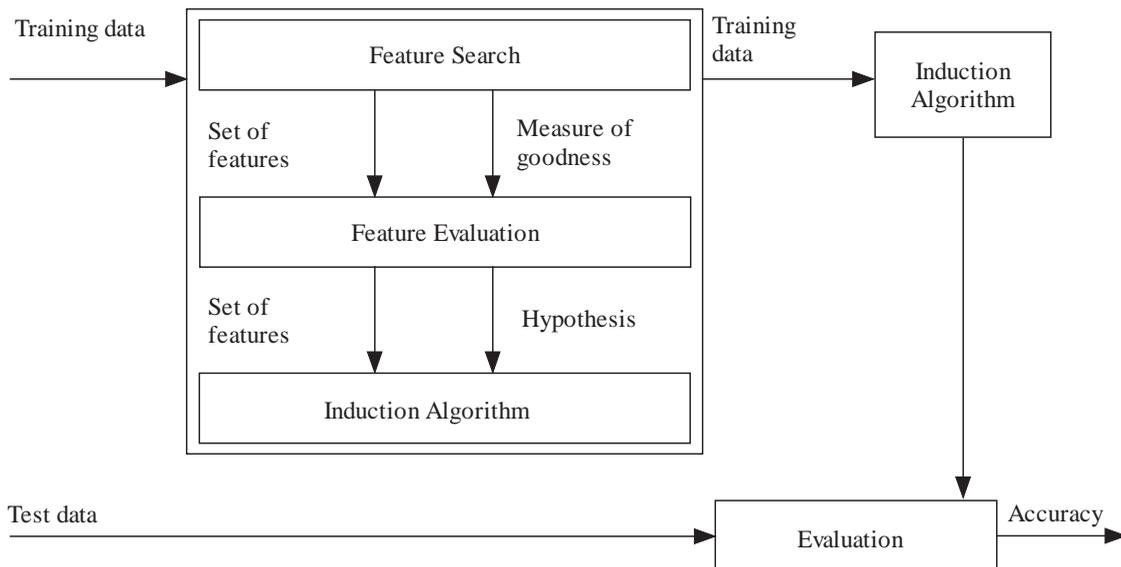
methods using support vector machines (SVMs) have obtained satisfactory results (Weston et al., 2001). SVMs are also combined with other techniques to implement feature selection (different approaches are described in Guyon et al., 2006). Kim et al. (2003) use artificial neural networks (ANNs) for customer prediction and ELSA (Evolutionary Local Selection Algorithm) to search for promising subsets of features.

Hybrid Methods

Whereas the computational cost associated with the wrapper model makes it unfeasible when the number of features is high, when the filter model is used its performance is less than satisfactory. The hybrid model is a good combination of the two approaches that overcomes these problems. Hybrid methods use a filter to generate a ranked list of features. On the basis of the order thus defined, nested subsets of features are generated and computed by a learning machine, i.e. following a wrapper approach (Guyon et al., 2006). The main features of the hybrid model are depicted in Figure 3. One of the first hybrid approaches proposed was that of Yuan et al., 1999. Since then, the hybrid model has focused the attention of the research community and, by now, numerous hybrid models have been developed to solve a variety of problems, such as intrusion detection, text categorisation, etc.

As a combination of filter and wrapper models, there exist a great number of hybrid methods, so it is not possible to include all of them and therefore we will refer to some interesting ones. Some hybrid methods involving SVMs are presented in Guyon et al. (2006), chapters 20 and 22. Shazzad & Park (2005) investigate a fast hybrid method –a fusion of Correlation-based Feature Selection, Support Vector Machine and Genetic Algorithm– to determine an optimal feature set. A feature selection model based both on information theory and statistical tests is presented by Sebban & Nock (2002). Zhu et al. (2007) incorporates a filter ranking method in a genetic algorithm to improve classification performance and accelerate the search process.

FUTURE TRENDS

Feature selection is a huge topic that it is impossible to discuss in a short chapter. To pinpoint new topics in this area we refer the reader to the suggestions given by Guyon et al. (2006), summarised as follows:

* Unsupervised variable selection. Although this chapter has focused on supervised feature selection, several authors have attempted to implement feature selection for clustering applications (see, for example, Dy & Brodley, 2004). For supervised learning tasks, one may want to pre-filter a set of the most significant variables with respect to a criterion which does not make use of y to minimise the problem of over-fitting.

Figure 3. Hybrid algorithm

F
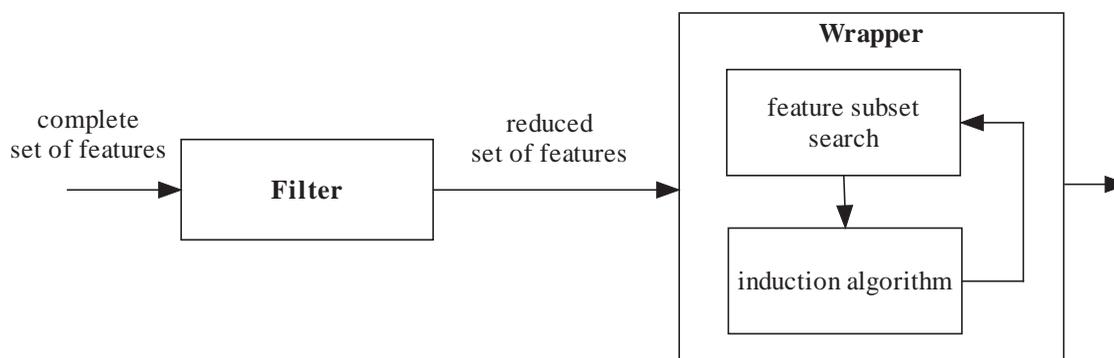
- Selection of examples. Mislabelled examples may induce a choice of wrong variables, so it may be preferable to jointly select both variables and examples.
- System reverse engineering. This chapter focuses on the problem of selecting features useful to build a good predictor. Unravelling the causal dependencies between variables and reverse engineering the system that produced the data is a far more challenging task that is beyond the scope of this chapter (but see, for example, Pearl, 2000).

CONCLUSION

Feature selection for classification and regression is a major research topic in machine learning. It covers many different fields, such as, for example, text categorisation, intrusion detection, and micro-array data. This study reviews key algorithms used for feature selection, including filter, wrapper and hybrid approaches. The review is not exhaustive and is merely designed to give an idea of the state of the art in the field. Most feature selection algorithms lead to significant reductions in the dimensionality of the data without sacrificing the performance of the resulting models. Choosing between approaches depends on the problem in hand. Adopting a filtering approach is computationally acceptable, but the more complex wrapper approach tends to produce greater accuracy in the final result. The filtering approach is very flexible, since any target learning algorithm can be used. It is also faster than the wrapper approach. This latter, on the other hand, is more dependent on the learning algorithm; but the selection process is better. The hybrid approach offers promise in terms of improving results in terms of classification accuracy as well as in terms of the identification of relevant attributes for the analysis.

REFERENCES

Aha, D.W., and Bankert, R. L. (1995). A comparative evaluation of sequential feature selection algorithms. *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, 1-7. Springer-Verlag.

Almuallim, H. & Dietterich, T. G (1991). Learning with many irrelevant features. *Proceedings of the 9th National Conference on Artificial Intelligence*, 547-552, AAAI Press.

Almuallim, H. & Dietterich, T. G. (1992) Efficient algorithms for identifying relevant features. *Proceedings of the 9th Canadian Conference on Artificial Intelligence*, 38-45, Vancouver.

Alpaydin, E. (2004). Introduction to Machine Learning. MIT Press.

Blum, A. L. & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, (97) 1-2, 245-271.

Caruana, R. & Freitag, D. (1994). Greedy attribute selection. *Proceedings of the Eleventh International Conference on Machine Learning.* Morgan Kaufmann Publishers, Inc., 28-36.

Dy, J. G. & Brodley, C. E. (2004). Feature Selection for Unsupervised Learning. *Journal of Machine Learning Research*, (5), 845–889.

Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, (3), 1157-1182.

Guyon, I., Gunn, S., Nikravesh, M. & Zadeh, L.A. (2006). Feature Extraction. Foundations and Applications. Springer.

Huang, J., Cai, Y. & Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Patter recognition letters,* (28) 13, 1825-1844.

Kim, Y., Street W. N. & Menczer, F. (2003). Feature selection in data mining. *Data mining: opportunities and challenges*, 80-105. IGI Publishing.

Kira, K. & Rendell, L. (1992). The feature selection problem: traditional methods and new algorithm. *Proc. AAAI'92*, San Jose, CA.

Kohavi, R. & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, (97)1-2, 273-324.

Liu, H. & Motoda, H. (1998). Feature extraction, construction and selection. A data mining perspective. Kluwer Academic Publishers.

Pearl, J. (2000). Casuality . Cambridge University Press.

Pudil, P. and Novovicova, J. and Kittler, J. (1994). Floating search methods in feature-selection. *Pattern Recognition Letters*, (15) 11, 1119-1125.

Robnik-Sikonja, M. & Kononenko, I. (2003). Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, (53), 23-69, Kluwer Academic Publishers.

Salappa, A., Doumpos, M. & Zopounidis, C. (2007). Feature selection algorithms in classification problems: an experimental evaluation. Optimization Methods and Software, (22) 1, 199 – 212.

Sánchez-Maroño, N., Caamaño-Fernández, M., Castillo, E & Alonso-Betanzos, A.(2006). Functional networks and analysis of variance for feature selection. *Proceedings of International Conference on Intelligent Data Engineering and Automated Learning,* 1031-1038.

Shazzad, K.M & Jong S.P. (2005). Optimization of Intrusion Detection through Fast Hybrid Feature Selection. *International Conference on Parallel and Distributed Computing, Applications and Technologies*, 264 – 267.

Sebban, M., Nock, R. (2002). A hybrid filter/wrapper approach of feature selection using information theory. *Patter recognition*, (35)4:835-846.

 Yu, L. and Liu, H. (2003). Feature selection for high-dimensional data: A Fast Correlation-Based Filter Solution. *Proceedings of The Twentieth International Conference on Machine Learning*, 856-863.

Yuan, H., Tseng, S.S., Gangshan, S. and Fuyan, Z. (1999). Two-phase feature selection method using both filter and wrapper. *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, (2) 132–136.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. (2001). Feature selection for SVMs. *Advances in Neural Information Processing Systems*, (13). MIT Press.

Zhao, Z. and Liu, H. (2007). Searching for interacting features.*Proceedings of International Joint Conference on Artificial Intelligence*, 1157-1161.

Zhu, Z., Ong, Y., Dash, M. (2007) Wrapper–Filter Feature Selection Algorithm Using a Memetic Framework. *IEEE Transactions on Systems, Man and Cybernetics*, Part B. (37) 1, 70-76.

KEY TERMS

Dimensionality Reduction: The process of reducing the number of features under consideration. The process can be classified in terms of feature selection and feature extraction.

Feature Extraction: A dimensionality reduction method that finds a reduced set of features that are a combination of the original ones.

Feature Selection: A dimensionality reduction method that consists of selecting a subset of relevant features from a complete set while ignoring the remaining features.

Filter Method: A feature selection method that relies on the general characteristics of the training data to select and discard features. Different measures can be employed: distance between classes, entropy, etc.

Hybrid Method: A feature selection method that combines the advantages of wrappers and filters methods to deal with high dimensionality data.

Sequential Backward (Forward) Selection (SBS/SFS): A search method that starts with all the features (an empty set of features) and removes (adds) a single feature at each step with a view to improving -or minimally degrading- the cost function.

Wrapper Method: A feature selection method that uses a learning machine as a "black box" to score subsets of features according to their predictive value.

Feed–Forward Artificial Neural Network Basics

Lluís A. Belanche Muñoz
Universitat Politècnica de Catalunya, Spain

The answer to the theoretical question: "Can a machine be built capable of doing what the brain does?" is yes, provided you specify in a finite and unambiguous way what the brain does.

Warren S. McCulloch

INTRODUCTION

The class of adaptive systems known as Artificial Neural Networks (ANN) was motivated by the amazing parallel processing capabilities of biological brains (especially the human brain). The main driving force was to re-create these abilities by constructing artificial models of the biological neuron. The power of biological neural structures stems from the enormous number of highly interconnected simple units. The simplicity comes from the fact that, once the complex electro-chemical processes are abstracted, the resulting computation turns out to be conceptually very simple.

These artificial neurons have nowadays little in common with their biological counterpart in the ANN paradigm. Rather, they are primarily used as *computational devices*, clearly intended to problem solving: optimization, function approximation, classification, time-series prediction and others. In practice few elements are connected and their connectivity is low. This chapter is focused to supervised feed-forward networks. The field has become so vast that a complete and clear-cut description of all the approaches is an enormous undertaking; we refer the reader to (Fiesler & Beale, 1997) for a comprehensive exposition.

BACKGROUND

Artificial Neural Networks (Bishop, 1995), (Haykin, 1994), (Hertz, Krogh & Palmer, 1991), (Hecht-Nielsen, 1990) are information processing structures without global or shared memory, where each of the computing elements operates only when all its incoming information is available, a kind of data-flow architectures.

Each element is a simple processor with internal and adjustable parameters. The interest in ANN is primarily related to the finding of satisfactory solutions for problems cast as function approximation tasks and for which there is scarce or null knowledge about the process itself, but a (limited) access to examples of response. They have been widely and most fruitfully used in a variety of applications—see (Fiesler & Beale, 1997) for a comprehensive review—especially after the boosting works of (Hopfield, 1982), (Rumelhart, Hinton & Williams, 1986) and (Fukushima, 1980).

The most general form for an ANN is a *labelled directed graph*, where each of the nodes (called *units* or *neurons*) has a certain computing ability and is connected to and from other nodes in the network via labelled edges. The edge label is a real number expressing the strength with which the two involved units are connected. These labels are called *weights*. The *architecture* of a network refers to the number of units, their arrangement and connectivity.

In its basic form, the computation of a unit i is expressed as a function F_i of its input (the *transfer* function), parameterized with its weight vector or local information. The whole system is thus a collection of interconnected elements, and the transfer function performed by a single one (i.e., the *neuron model*) is the most important fixed characteristic of the system.

There are two basic types of neuron models in the literature used in practice. Both express the overall computation of the unit as the composition of two functions, as is classically done since the earlier model proposal of McCulloch & Pitts (1943):

$$F_i(x) = \{g(h(x,w_i)), \ w_i \in R^n\}, \quad x \in R^n \quad (1)$$

where w_i is the weight vector of neuron i, $h: R^n \times R^n \rightarrow R$ is called the *net input* or *aggregation* function, and $g: R \rightarrow R$ is called the *activation* function. All neuron parameters are included in its weight vector.

The choice $h(x,w_i) = x \cdot w_i + \theta$, where $\theta \in R$ is an offset term that may be included in the weight vector, leads to one of the most widely used neuron models. When

Figure 1. A classification problem. Left: Separation by spherical RBF units (R-neurons). Right: Separation by straight lines (P-neurons) in the MLP.
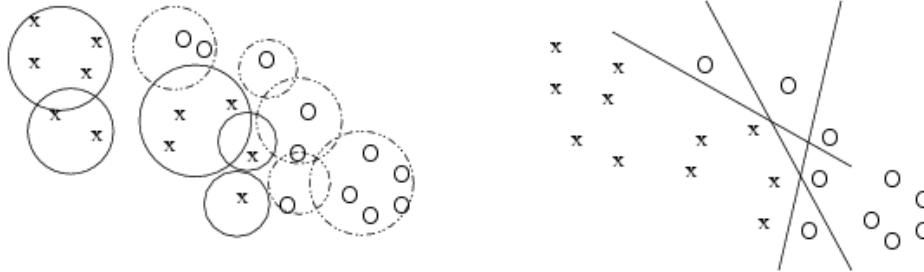
neurons of this type are arranged in a feed-forward architecture, the obtained neural network is called MultiLayer Perceptron (MLP) (Rumelhart, Hinton & Williams, 1986). Usually, a smooth non-linear and monotonic function is used as *activation*. Among them, the sigmoids are a preferred choice.

The choice $h(\boldsymbol{x},\boldsymbol{w}_i) = ||\boldsymbol{x}-\boldsymbol{w}_i||/\theta$ (or other distance measure), with $\theta > 0 \in R$ a smoothing term, plus an activation g with a monotonically decreasing response from the origin, leads to the wide family of localized Radial Basis Function networks (RBF) (Poggio & Girosi, 1989). Localized means that the units give a significant response only in a neighbourhood of their centre $\boldsymbol{w}_i$. A Gaussian $g(z)=exp(-z^2/2)$ is a preferred choice for the activation function.

The previous choices can be extended to take into account extra correlations between input variables. The inner product (containing no cross-product terms) can be generalized to a real quadratic form (an homogeneous polynomial of second degree with real coefficients) or even further to higher degrees, leading to the so-called higher-order units (or $\Sigma-\Pi$ units). A higher-order unit of degree k includes all possible cross-products of at most k input variables, each with its own weight. Conversely, basic Euclidean distances can be generalized to completely weighted distance measures, where all the (quadratic) cross-products are included. These full expressions are not commonly used because of the high numbers of free parameters they involve.

These two basic neuron models have traditionally been regarded as completely separated, both from a mathematical and a conceptual point of view. To a certain degree, this is true: the local vs. global approximation approaches to a function that they carry out make them apparently quite opposite methods (see Fig. 1). Mathematically, under certain conditions, they can be shown to be related (Dorffner, 1995). These conditions (basically, that *both* input and weight vectors are normalized to unit norm) are difficult to fulfil in practice.

A *layer* is defined as a collection of independent units (not connected with one another) sharing the same input, and of the same functional form (same F_i but different $\boldsymbol{w}_i$). Multilayer feed-forward networks take the form of directed acyclic graphs obtained by concatenation of a number of layers. All the layers but the last (called the output layer) are labelled as *hidden*. This kind of networks (shown in Fig. 2) compute a parameterized function $F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})$ of their input vector $\boldsymbol{x}$ by evaluating the layers in order, giving as final outcome the output of the last layer. The vector $\underline{\boldsymbol{w}}$ represents the collection of all the weights (free parameters) in the network. For simplicity, we are not considering connections between non-adjacent layers (*skip-layer connections*) and assume otherwise total connectivity. The set of input variables is *not* counted as a layer.

Output neurons take the form of a scalar product (a linear combination), eventually followed by an activation function g. For example, assuming a single output neuron, a one-hidden-layer neural network with h hidden units computes a function $F:R^n \rightarrow R$ of the form:

$$F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})=g(\sum_{i=1}^{h} c_i F_i(\boldsymbol{x}) - \theta) \qquad (2)$$

Figure 2. A two-hidden-layer example of ANN, mapping a three-dimensional input space x=(x1,x2,x3) to a two-dimensional output space (y1,y2)=Fw(x). The network has four and three units in the first and second hidden layers, respectively, and two output neurons. The vector w represents the collection of all the weights in the network.
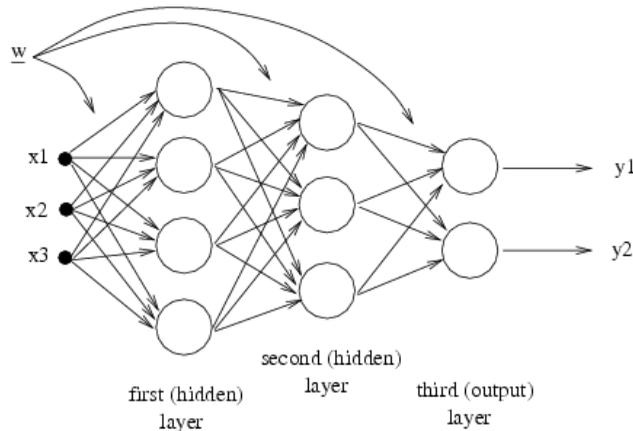
where $\theta \in R$ is an offset term (called the *bias* term), $c_i \in R$ and g can be set as desired, including the choice $g(z)=z$. Such a feed-forward network has $dim(\underline{w})=(n+1)h+h+1$ parameters to be adjusted.

FEED-FORWARD NEURAL NETWORKS

The RBF and MLP networks provide parameterized families of functions suitable to function approximation on multidimensional spaces. A sigmoid neuron puts up an hyperplane that divides its input space in two halves. In other words, the points of equal neuron activation (with fixed weights) are hyperplanes. This behaviour is not caused by the sigmoid, but by the scalar product. The isoactivation contours for an RBF unit (in case of an unweighted Euclidean norm) are hyperspheres. The radially symmetric and centered response is not caused by the activation function (e.g., Gaussian or exponential) but by the norm. In both cases, the activation function acts as a non-linear monotonic distorsion of its argument as computed by the aggregation function.

Definition (Isoactivation set). Given a real function $f:R^n \rightarrow (a,b)$, define I_f^α for $\alpha \in (a,b)$ as the set of isoactivation points $I_f^\alpha = \{x \in R^n | f(x)=\alpha\}$.

Definition (P-neuron). A neuron model F_i of the form:

$$F_i(x)=\{g(w_i \cdot x + \theta_i), w_i \in R^n, \theta_i \in R\} \qquad (3)$$

with g a bounded, non-linear and increasing function for which $\lim_{z \to \infty} g(z)=g_{max} \in R$ and $\lim_{z \to -\infty} g(z)=g_{min} \in R$ will be denoted *P*-neuron (from Perceptron). For these neurons, the sets I_{Fi}^α are $(n$-1)-dimensional hyperplanes for constant values of α, parallel with one another for different α. In practice, the g are usually the well-behaved sigmoids, though other activation functions are sometimes found in the literature (e.g., sinusoid). The latter are not included in the above Definition.

Definition (R-neuron). A neuron model F_i of the form:

$$F_i(x)=\{1/\theta_i \, g(||x-w_i||_q), \; w_i \in R^n, \; \theta_i >0 \in R, \; q \geq 1 \in R\} \qquad (4)$$

where $|.|$ *is a norm* and g is a symmetric function such that $g(/z/)$ is monotonic, with a maximum g_{max} at $F_i(w_i)$ and a (possibly asymptotically reached) minimum $g_{min}=0$ will be denoted *R*-neuron (from Radial). For these neurons, the sets I_{Fi}^α are $(n$-1)-dimensional

Figure 3. The logistic function $l(z)=g^{log}{}_{1.5}(z)$ and its first derivative $l'(z)>0$. This function is maximum at the origin, corresponding to a medium activation at $l(0)=0.5$. This point acts as an initial "neutral" value around a quasi-linear slope.
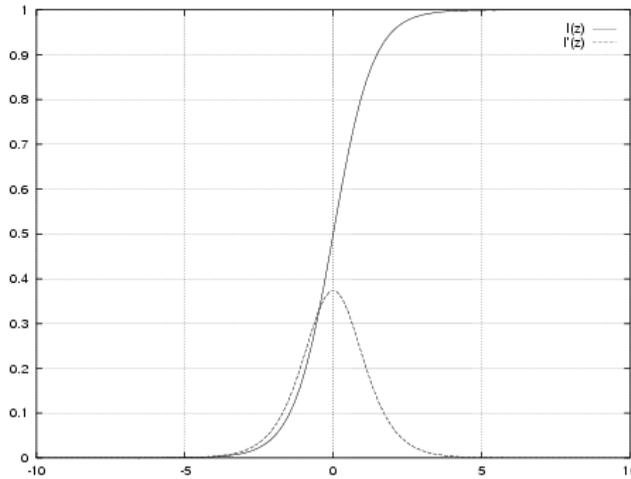
hypersurfaces (centered at w_i) for constant values of α (e.g., hypercubes for $q=1$, hyperspheres for $q=2$) concentric with one another for different α.

The norm used can be any Minkowskian norm of the form:

$$\|z\|_q = (\sum_{i=1}^{n} |z_i|^q)^{1/q}, \quad q \geq 1 \in R \qquad (5)$$

In practice, typical choices are $q=2$ and g a Gaussian function.

Due to their widespread use, we present two of the most popular sigmoids, and show how they are tightly related. A *sigmoid* function g can be defined as a monotonically increasing function exhibiting smoothness and asymptotic properties. The two more commonly found representatives are the **logistic**:

$$\frac{1}{1+\exp(-\beta(z-\theta))} \in (0,1) \qquad (6)$$

and the **hyperbolic tangent**:

$$g^{tanh}{}_{\beta}(z) = \frac{\exp(\beta(z-\theta)) - \exp(-\beta(z-\theta))}{\exp(\beta(z-\theta)) + \exp(-\beta(z-\theta))} \in (-1,1)$$

$$(7)$$

The offset θ is in practice set to zero, because its function is the same as that of the *bias* term in the aggregation function in (3). These two families of functions can be made exactly the same shape (assuming $\theta=0$) by making the β in (6) be twice the value of the β in (7). For instance, for $\beta=0.5$:

$$g^{tanh}{}_{0.5}(z) = g^{tanh}{}_{1}(z/2) = \frac{1-\exp(-z)}{1+\exp(-z)} = 2g^{log}{}_{1}(z)\text{-}1$$

$$(8)$$

is the bipolar version of $g^{log}{}_{1}(z) = \frac{1}{1+\exp(-z)}$. These functions are chosen because of their simple analytic behaviour, especially in what concerns differentiability, of great importance for learning algorithms relying in derivative information (Fletcher, 1980). In particular,

$$(g^{log}{}_{\beta})'(z) = \beta \, g^{log}{}_{\beta}(z)(1\text{-}g^{log}{}_{\beta}(z)) \qquad (9)$$

The interest in sigmoid functions also relies in the behaviour of their derivatives. Consider, for example, (6) with $\beta=1.5$ and $\theta=0$, plotted in Fig. (3). The derivative of a sigmoid is always positive. For $\theta=0$, all the functions are centred at $z=0$. In this point, the function has a medium activation, and its derivative is maximum, allowing for maximum weight updates.

Types of Artificial Neural Networks

A fundamental distinction to categorize a neural network relies on the kind of architecture, basically divided in *feed-forward* (for which the graph contains no cycles) and *recurrent* (the rest of situations). A very common feed-forward architecture contains no intra-layer connections and all possible inter-layer connections between adjacent layers.

Definition (Feed-forward neural network: structure). A *bipartitioned graph* is a graph G whose nodes V can be partitioned in two disjoint and proper sets V_1 and V_2, $V_1 \cup V_2 = V$, in such a way that no pair of nodes in V_1 is joined by an edge, and the same property holds for V_2. We write then $G_{n1,n2}$, with $n_1=/V_1/,n_2=/V_2/$. A bipartitioned graph $G_{n1,n2}$ is *complete* if every node in V_1 is connected to every node in V_2. These concepts can be generalized to an arbitrary number of partitions, as follows: A k-partitioned graph $G_{n1,...,nk}$ is a graph whose nodes V can be partitioned in k disjoint and proper sets $V_1,...,V_k$, such that

$$\bigcup_{i=1}^{k} V_i = V,$$

in a way that no pair of nodes in V_i is joined by an edge, for all $1 \le i \le k$. In these conditions, a feed-forward fully connected neural network with c hidden layers and h_l units per layer l, $1 \le l \le c+1$, takes the form of a directed complete $c+1$-partitioned graph $G_{h1,...,hc+1}$.

Definition (Feed-forward neural network: function). A feed-forward neural network consisting of c hidden layers, denoted FFNN(n,c,m), is a function $F_w: R^n \to R^m$ made up of pieces of the form $y^{(l)}=(F_1^l(y^{(l-1)}),...,F_{hl}^l(y^{(l-1)}))$, representing the output of layer l, for $1 \le l \le c+1$. The F^l denote the neuron model of layer l and $h_l \in N^+$ their number, and each neuron F_i^l has its own parameters $w^{(l)}_i$ as

in Definitions 2 and 3, which are collectively grouped in the *network parameters w*. The first output is defined as $y^{(0)}=x$. For the last (output) layer, $h_{c+1}=m$ and the $F^{c+1}_l, 1 \le l \le h_{c+1}$ are P-neurons or *linear* units (obtained by removing the activation function in a P-neuron). The final outcome for $F_w(x)$ is the value of $y^{(c+1)}$.

Definition (MLPNN). A MultiLayer Perceptron Neural Network is a FFNN (n,c,m) for which $c\ge1$ and all the F^l are P-neurons, $1 \le l \le c$.

Definition (RBFNN). A Radial Basis Function Neural Network is a FFNN (n,c,m) for which $c=1$ and all the F^c are R-neurons.

LEARNING IN ARTIFICIAL NEURAL NETWORKS

A system can be said to *learn* if its performance on a given task improves with respect to some measure as a result of experience (Rosenblatt, 1962). In ANNs the "experience" is the result of exposure to a training set of data, accompanied with weight modifications. The main problem tackled in *supervised learning* is regression, the approximation of an n-dimensional function $f: X \subset R^n \to R^m$ by finite superposition (composition and addition) of known parameterized *base* functions, like those in (3) or (4). Their combination gives rise to expressions of the form $F_w(x)$. The interest is in finding a parameter vector w^* of size s such that $F_{w^*}(x)$ optimizes a cost functional L (f, F_w) called the *loss*:

$$w^*=argmin_{w \in R^s} \text{ L} (f, F_w) \qquad (10)$$

The only information available is a finite set D of p noisy samples of f, $D=\{<x_i,y_i>,f(x_i)+\varepsilon_i=y_i\}$, where $x_i \in R^n$ is the stimulus, $y_i \in R^m$ is the target, ε_i is the noise (assumed additive) and $/D/=p$. An estimation of L (f, F_w) can be obtained as $\tilde{L}(D, F_w)$, the *apparent loss*, computed separately for each sample in D,

$$\tilde{L}(D, F_w) = \sum_{(x_i,y_i) \in D} \lambda(y_i, F_w(x_i)) \qquad (11)$$

A common form for λ is an error function, as the squared-error $\lambda(a,b)=(a-b)^2$. This results from the assumption that the noise follows a homocedastic gaussian distribution with zero mean. When using this error, the expression (11) can be viewed as the (squared) Euclidean norm in R^p of the p-dimensional error vector $\boldsymbol{e}=(e_1,...,e_p)$, known as the sum-of-squares error, with $e_i=y_i-F_{\underline{\boldsymbol{w}}}(\boldsymbol{x}_i)$, as:

$$\tilde{L}(D,F_{\underline{\boldsymbol{w}}}) = \sum_{(\mathbf{x}_i,\mathbf{y}_i)\in D}(y_i-F_{\underline{\boldsymbol{w}}}(\boldsymbol{x}_i))^2 = \boldsymbol{e}\cdot\boldsymbol{e} = ||\boldsymbol{e}||^2$$

$$(12)$$

The usually reported quantity $||\boldsymbol{e}||^2/p$ is called *mean square error* (MSE), and is a measure of the empirical error (as opposed to the unknown *true* error). We shall denote the error function $E(\underline{\boldsymbol{w}})=\tilde{L}(D,F_{\underline{\boldsymbol{w}}})$. In a training process, the network builds an internal representation of the target function by finding ways to combine the set of base functions $\{F_i(\boldsymbol{x})\}_i$. The validity of a solution is mainly determined by an acceptably low and balanced $\tilde{L}(D,F_{\underline{\boldsymbol{w}}})$ and $\tilde{L}(D_{out},F_{\underline{\boldsymbol{w}}})$, for any $D_{out}\subset X\backslash D$ (where D_{out} is not used in the learning process) to ensure that f has been correctly estimated from the data. Network models too inflexible or simple or, on the contrary, too flexible or complex will generalize inadequately. This is reflected in the bias-variance tradeoff: the expected loss for finite samples can be decomposed in two opposing terms called *error bias* and *error variance* (Geman, Bienenstock & Doursat, 1992). The expectation for the sum-of-squares error function, averaged over the complete ensemble of data sets D is written as (Bishop, 1995):

$$E(\underline{\boldsymbol{w}}) = E_D\{(F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})-<y/\boldsymbol{x}>)^2\}$$

$$= (E_D\{(F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})-<y/\boldsymbol{x}>)\})^2 + E_D\{(F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})-E_D\{F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})\})^2\}$$

$$(13)$$

where E_D is the expectation operator taken over every data set of the same size as D and $<y/\boldsymbol{x}>$ denotes the

conditional average of the target $y=f(\boldsymbol{x})$ (which expresses the optimal network mapping), given by:

$$<y/\boldsymbol{x}>=\int yp(y/\boldsymbol{x})dy$$

The first term in the right hand side of (13) is the (squared) bias and the second is the variance. The bias measures the extent to which the average (over all D) of $F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})$ differs from the desired target function $<y/\boldsymbol{x}>$. The variance measures the sensitivity of $F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})$ to the particular choice of D. Too inflexible or simple models will have a large bias, while too flexible or complex will have a large variance. These are complementary quantities that have to be minimized simultaneously; both can be shown to decrease with increasing availability of larger data sets D.

The expressions in (13) are functions of an input vector $\boldsymbol{x}$. The average values for bias and variance can be obtained by weighting with the corresponding density $p(\boldsymbol{x})$:

$$\int E_D\{(F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})-<y/\boldsymbol{x}>)^2\}p(\boldsymbol{x})d\boldsymbol{x}$$

$$= \int (E_D\{(F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})-<y/\boldsymbol{x}>)\})^2p(\boldsymbol{x})d\boldsymbol{x}$$

$$+ \int E_D\{(F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})-E_D\{F_{\underline{\boldsymbol{w}}}(\boldsymbol{x})\})^2\}p(\boldsymbol{x})d\boldsymbol{x}$$

$$(14)$$

Key conditions for acceptable performance on novel data are given by a training set D as large and representative as possible of the underlying distribution, and a set D_{out} of previously unseen data which should not contain examples exceedingly different from those in D. An important consideration is the use of a net with minimal complexity, given by the number of free parameters (the number of components in $\underline{\boldsymbol{w}}$). This requirement can be realized in various ways. In regularization theory, the solution is obtained from a variational principle including the loss and prior smoothness information, defining a *smoothing functional* ϕ such that lower values correspond to smoother functions. A solution of the approximation problem is then given by minimization of the functional (Girosi, Jones & Poggio, 1993):

$$H(F_{\underline{w}}) = \tilde{L}\,(D, F_{\underline{w}}) + \eta\phi(F_{\underline{w}}) \qquad (15)$$

where η is a positive scalar controlling the tradeoff between fitness to the data and smoothness of the solution. A common choice is the second derivative $P(f) = f''$ of which the (squared) Euclidean norm is taken:

$$\phi(F_{\underline{w}}) = //P(F_{\underline{w}})//^2 = \int \{F_{\underline{w}}''(t)\}^2 dt \qquad (16)$$

CONCLUSION

Artificial Neural Networks are information processing structures evolved as an abstraction of known principles of how the brain might work. The computing elements, called *neurons*, are linked to one another with a certain strength, called *weight*. In their simplest form, each unit computes a *function* of its inputs—which are either the outputs from other units or external signals—influenced by the weights of the links conveying these inputs. The network is said to *learn* when the weights of all the units are adapted to represent the information present in a sample, in an optimal sense given by an error function. The network relies upon the representation capacity of the neuron model as the cornerstone for a good approximation.

REFERENCES

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.

Dorffner, G. (1995). A generalized view on learning in feedforward neural networks. Technische Universität Cottbus, Reihe Mathematik M-01/1995, pp.34-54.

Fiesler, E., Beale, R. (Eds., 1997) *Handbook of Neural Computation*. IOP Publishing & Oxford Univ. Press.

Fletcher, R. (1980). *Practical methods of optimization*. Wiley.

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, pp. 193-202.

Geman, S., Bienenstock, E., Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4 (1): 1-58, 1992.

Girosi, F., Jones, M., Poggio, T. (1993). Priors, Stabilizers and Basis Functions: from regularization to radial, tensor and additive splines. AI Memo No.1430, AI Laboratory, MIT.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. MacMillan.

Hecht-Nielsen, R. (1990). *Neurocomputing*. Addison-Wesley.

Hertz, J., Krogh, A., Palmer R.G. (1991). *Introduction to the Theory of Neural Computation*, Addison-Wesley.

Hopfield, J.J. (1982) Neural Networks and Physical Systems with Emergent Collective and Computational Abilities. In Proceedings of the National Academy of Sciences, USA, Vol. 79, pp. 2554-2558.

McCulloch, W., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5: 115-133.

Poggio T., Girosi, F. (1989). A Theory of Networks for Approximation and Learning. AI Memo No. 1140, AI Laboratory, MIT.

Rosenblatt, F. (1962). *Principles of neurodynamics*. Spartan Books, NY.

Rumelhart, D., Hinton, G., Williams, R. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1: Foundations). Rumelhart, McClelland (eds.), MIT Press, Cambridge, MA.

KEY TERMS

Architecture: The number of artificial neurons, its arrangement and connectivity.

Artificial Neural Network: Information processing structure without global or shared memory that takes the form of a directed graph where each of the computing elements ("neurons") is a simple processor with internal and adjustable parameters, that operates only when all its incoming information is available.

F

Bias-Variance Tradeoff: The mean square error (to be minimized) decomposes in a sum of two non-negative terms, the squared bias and the variance. When an estimator is modified so that one term decreases, the other term will typically increase.

Feed-Forward Artificial Neural Network: Artificial Neural Network whose graph has no cycles.

Learning Algorithm: Method or algorithm by virtue of which an Artificial Neural Network develops a representation of the information present in the learning examples, by modification of the weights.

Neuron Model: The computation of an artificial neuron, expressed as a function of its input and its weight vector and other local information.

Weight: A free parameter of an Artificial Neural Network, that can be modified through the action of a Learning Algorithm to obtain desired responses to input stimuli.

Finding Multiple Solutions with GA in Multimodal Problems

Marcos Gestal
University of A Coruña, Spain

Mari Paz Gómez-Carracedo
University of A Coruña, Spain

INTRODUCTION

Traditionally, the Evolutionary Computation (EC) techniques, and more specifically the Genetic Algorithms (GAs) (Goldberg & Wang, 1989), have proved to be efficient when solving various problems; however, as a possible lack, the GAs tend to provide a unique solution for the problem on which they are applied. Some non global solutions discarded during the search of the best one could be acceptable under certain circumstances. The majority of the problems at the real world involve a search space with one or more global solutions and multiple local solutions; this means that they are multimodal problems (Harik, 1995) and therefore, if it is desired to obtain multiple solutions by using GAs, it would be necessary to modify their classic functioning outline for adapting them correctly to the multimodality of such problems.

MOTIVATION

This chapter tries to establish the basis for the understanding of multimodality where, firstly, the characterisation of the multimodal problems will be attempted. It would be also tried to offer a global view of some of the several approaches proposed for adapting the classic functioning of the GAs to the search of multiple solutions. Lastly, the contributions of the authors will be also showed.

BACKGROUND: CHARACTERIZATION OF MULTIMODAL PROBLEMS

The multimodal problems can be briefly defined as those problems that have multiple global optimums or multiple local optimums.

For this type of problems, it is interesting to obtain the greatest number of solutions due to several reasons; on one hand, when there is not a total knowledge of the

Figure 1. Rastrigin function
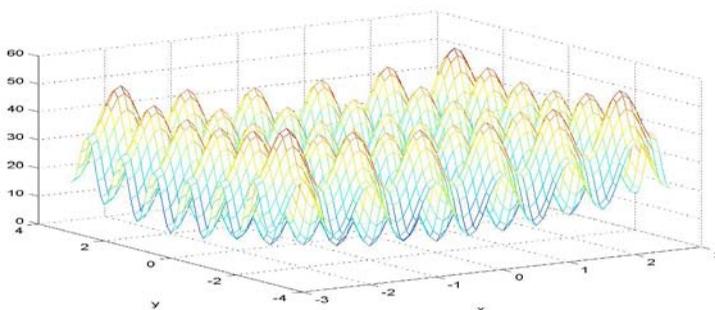
problem, the solution obtained might not be the best one as it can not be stated that no better solution could be found at the search space that has not been explored yet. On the other hand, although being certain that the best solution has been achieved, there might be other equally fitted or slightly worst solutions that might be preferred due to different factors (easier application, simpler interpretation, etc.) and therefore considered globally better.

One of the most characteristic multimodal functions used in lab problems are the Rastrigin function (see Fig. 1) which offers an excellent graphical point of view about multimodality means.

Providing multiple optimal (and valid) solutions and not only the unique global solution is crucial in multiple environments. Usually, it is very complex to implement in the practice the best solution represents, so it can offers multiple problems: computational cost too high, complex interpretation,…

In these situations it turns out useful to have a range of valid solutions between which that one could choose that, still not being the best solution to the raised problem, offer a level of acceptable adjustment and be simpler to implement, to understand, … that the ideal global one.

EVOLUTIONARY TECHNIQUES AND MULTIMODAL PROBLEMS

As it has been mentioned, the application of EC techniques to the resolution of multimodal problems sets out the difficulty that this type of techniques shows since they tend to solely provide the best of the found solutions and to discard possible local optimums that might have been found throughout the search. Quite many modifications have been included in the traditional performance of the GA in order to achieve good results with multimodal problems.

A crucial aspect when obtaining multiple solutions consists on keeping the diversity of the genetic population, distributing as much as possible the genetic individuals throughout the search space.

CLASSICAL APPROACHES

Nitching methods allow GAs to maintain a genetic population of diverse individuals, so it is possible to locate multiple optimal solutions within a single population.

In order to minimise the impact of homogenisation, or to tend that it may only affect later states of searching phase, several alternatives have been designed, based most of them on heuristics. One of the first alternatives for promoting the diversity was the applications of scaling methods to the population in order to emphasize the differences among the different individuals. Other direct route for avoiding the diversity loss involves focusing on the elimination of duplicate partial high fitness solutions (Bersano, 1997) (Langdon, 1996).

Some other of the approaches tries to solve this problem by means of the dynamic variation of crossover and mutation rates (Ursem, 2002). A higher amount of mutations are done in order to increase the exploration through the search space, when diversity decreases; the mutations decrease and crossovers increase with the aim of improving exploitation in optimal solution search when diversity increases. There are also proposals of new genetic operators or variations of the actual ones. For example some of the crossover algorithms that improve diversity and that should be highlighted are BLX (Blend Crossover) (Eshelman & Schaffer, 1993), SBX (Simulated Binary Crossover) (Deb & Agrawal, 1995), PCX (Parent Centric Crossover) (Deb, Anand & Joshi, 2002), CIXL2 (Confidence Interval Based Crossover using L2 Norm) (Ortiz, Hervás & García, 2005) or UNDX (Unimodal Normally Distributed Crossover) (Ono & Kobayashi, 1999).

Regarding replacement algorithms, schemes that may keep population diversity have been also looked for. An example of this type of schemes is crowding (DeJong, 1975)(Mengshoel & Goldberg, 1999). Here, a newly created individual is compared to a randomly chosen subset of the population and the most closely individual is selected for replacement. Crowding techniques are inspired by Nature where similar members in natural populations compete for limited resources. Likewise, dissimilar individuals tend to occupy different niches and are unlikely to compete for the same resource, so different solutions are provided.

Fitness sharing was firstly implemented by Goldberg & Richardson for being used on multimodal functions (Goldberg & Richardson, 1999). The basic idea involves determining, from the fitness of each solution, the maximum number of individuals that can remain around it, awarding the individuals that exploit unique areas of the domain. The dynamic fitness shar-

ing (Miller & Shaw, 1995) with two components was proposed in order to correct the dispersion of the final distribution of the individuals into niches: the distance function, which measures the overlapping of individuals, and the comparison function, which results "1" if the individuals are identical and values closer to "0" as much different they are.

The clearing method (Petrowski, 1996) is quite different from the previous ones, as the resources are not shared, but assigned to the best individuals, who will be then kept at every niche.

The main inconvenience of the techniques previously described lies in the fact that they add new parameters that should be configured according the process of execution of GA. This process may be disturbed by the interactions among those parameters (Ballester & Carter, 2003).

OWN PROPOSALS

Once detected the existing problems they should be resolved, or at least, minimized. With this goal, the Artificial Neural Network and Adaptive System (RNASA) group have developed two proposals that use EC techniques for this type of problems. Both proposals try to find the final solution but keeping partial solutions within the final population.

The main ideas of the two proposals, together with the problems used for the tests are explained at the following points.

Hybrid Two-Population Genetic Algorithm

Introduction

To force a homogeneous search throughout the search space, the approach proposed here is based on the addition of a new population (genetic pool) to a traditional GA (secondary population). The genetic pool will divide the search space into sub-regions. Every one of the individuals of the genetic pool has its own fenced range for gene variation, so every one of these individuals would represent a specific sub-region within the global search space. On the other hand, the group of individual ranges in which any gene may have its value, is extended over the whole of those possible values that a gene may have. Therefore, this genetic pool would sample the whole of the search space.

It should be borne in mind that a traditional GA performs its search considering only one sub-region (the whole of the search space). Here the search space will be divided into different subregions or intervals according to the number of genetic individuals in the genetic pool.

Since the individuals in the genetic pool have restrictions in their viable gene values, one of these individuals would not be provided a valid solution. So, it is also used another population (the secondary population) in addition to the genetic pool. Here, a classical GA would develop its individuals in an interactive fashion with those individuals of the genetic pool.

Unlike at genetic pool, the genes of individuals of secondary population may adopt values throughout the

Figure 2. Structure of populations of hybrid two-population genetic algorithm

Genetic Pool

Secondary Population

N: Number of Variables Selected   **P:** Genetic Pool Individuals
S: Secondary Population Individuals   **d:** number of original variables
n: number of subregions to divide the search space
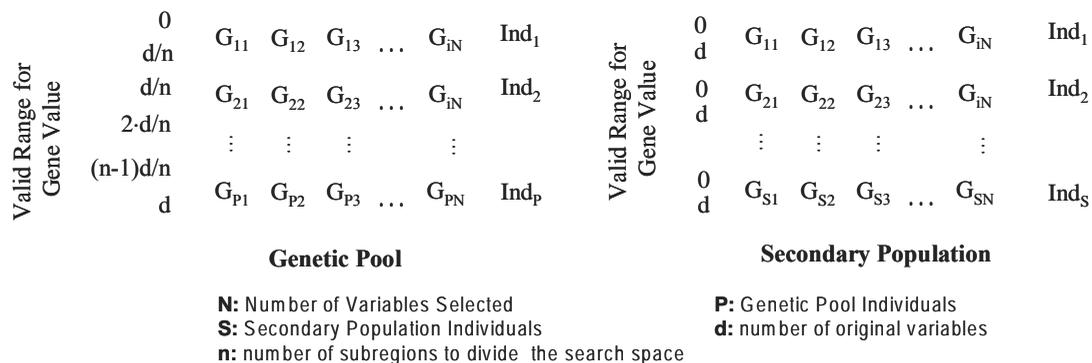
whole of the search space, so it would contribute the solutions, whereas the genetic pool would act as a support, keeping search space homogeneously explored.

The secondary population will provide the solutions (since its individuals are allowed to vary along all the search space range), whereas the genetic pool would act as a support, keeping search space homogeneously explored.

Next, both populations, which are graphically represented in Fig. 2, will be described in detail.

The Genetic Pool

As it has been previously mentioned, every one of the individuals at the genetic pool represents a sub-region of the global search space. Therefore, they should have the same structure or gene sequence than when using a traditional GA. The difference lies in the range of values that these genes might have.

When offering a solution, traditional GA may have any valid value, whereas in the proposed GA, the range of possible values is restricted. Total value range is divided into the same number of parts than individuals in genetic pool, so that a sub-range of values is allotted to each individual. Those values that a given gene may have will remain within its range for the whole of the performance of the proposed GA.

In addition to all that has been said, every individual at the genetic pool will be in control of which are the genes that correspond to the best found solution up to then (meaning whether they belong to the best individual at secondary population). This Boolean value would be used to avoid the modification of those genes that, in some given phase of performance, are the best solution to the problem.

Furthermore, every one of the genes in an individual has an I value associated which indicates the relative increment that would be applied to the gene during a mutation operation based only on increments and solely applied to individuals of the genetic pool. It is obvious that this incremental value should have to be

lower than the maximum range in which gene values may vary. The structure of the individuals at genetic pool is shown at Fig.3.

As these individuals do not represent global solutions to the problem that has to be solved, so their fitness value will not be compulsory. It will reduce the complexity of the algorithm and, of course, it will increase the computational efficiency of the final implementation.

The Secondary Population

The individuals of the secondary population are quite different for the previous. In this case, the genes of the individuals on the secondary population can take any value throughout the whole space of possible solutions. This allows that all individuals on secondary population are able to offer global solutions to the problem. This is not possible in genetic pool because their genes were restricted to different sub-ranges.

The evolution of the individuals at the genetic pool will be carried out by a traditional GA rules. The main different lies in the operator crossover. In this case a modified crossover will be used. Due to the information is stored in isolated population, now the two parents who will produce the new offspring will not belong to the same population. Hence, the genetic pool and secondary population are combined instead. In this way information of both populations will be merged to produce the most fitted offspring.

The Crossover Operator

As it was pointed before the crossover operator recombines the genetic material of the individuals of both populations. This recombination involves a random individual from secondary population with a representative of the genetic pool.

This representative will represent a potential solution offered by the genetic pool. As a unique individual can not verify this requirement, the representative will

Figure 3. Structure of the genetic pool individuals

Gene Values   **Control: Best Solution?**   **Mutation Increment**
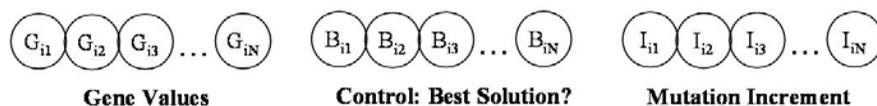
Figure 4. Hybrid two-population genetic algorithm: Crossover
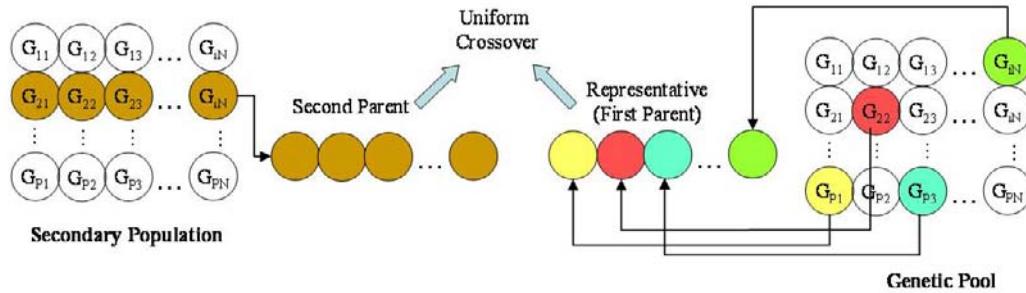
be formed by a subset of genes of different individuals on the genetic pool. Gathering information from different partial solutions will allow producing a valid global solution.

Therefore, the value for every gene of the representative will be randomly chosen among all the individuals in the genetic pool. After a value is assigned to all the genes, this new individual represents not a partial, unlike every one of the individuals separately, but a global solution.

Now, the crossover operator will be applied. This crossover function will keep the secondary population diversity, so the offspring will contain values from the genetic pool. Therefore the genetic algorithm would be able to maintain multiple solutions in the same population. The crossover operator does not change the genetic pool because the last one only acts as an engine to keep the diversity

This process is summarized in Fig 4.

The Mutation Operator

Mutation operator increments the value of individual genes in the genetic pool. It introduces new information in the genetic pool, so the representative can use it and finally, by means of the crossover operator, introduce it in secondary population.

It should be noted that the new value will have upper limit, so when it is reached the new gene value will be reset to the lower value.

When generations advance the increment amount is reduced, so the increment applied to the individuals in the genetic pool will take lower values. The different increments between iterations are calculated taking in mind the lower value for a gene (LIM_INF_IND), the

upper value for that gene (LIM_SUP_IND) and the total number of individuals in the genetic pool (IND_POOL) as Fig. 5 summarize. In such way, first generations will explore the search space briefly (a coarse-grain search) and it is intended to do a more exhaustive route through all the values that a given gene may have (a fine-grain search) as the search process advance.

Genetic Algorithm with Division into Species

Another proposed solution is an adaptation of the nitching technique. This adaptation consists on the division of the genetic population into different and independent subspecies. In this case the criterion that determines the specie for a specific individual to concrete specie is done according to genotype similarities (similar genotypes will form isolated species). This classical concept has been provided with some improvements in order to, not only decrease the number of iterations needed for obtaining solutions, but also increase the number of solutions kept within the genetic population.

Several iterations of the GA were executed on every species of the genetic population for speeding up the convergence towards the solution that exists near every species. The individuals generated during this execution having a genotype of a different species will be discarded.

The crossover operations between the species are following applied similarly to what happens in biology. It origins, on one hand, the crossovers between similar individuals are preferred (as it was done at the previous step using GAs) and on the other, the crossovers between different species are enabled, although in a lesser rate.

Figure 5. Pseudocode for mutation and Delta initialization

```
IF (not Bi)
   Gi = Gi + Ii
   IF (Gi > LIM_SUP_GEN)
     Gi = LIM_INF_GEN
     Ii = Ii – Delta
   ENDIF
ENDIF
```

$$\text{Delta} = \frac{(LIM\_SUP\_IND)-(LIM\_INF\_IND)}{IND\_POOL}$$

The individuals generated after these crossovers could, either be incorporated to an already existing species or, if they analyse a new area of the search space, create themselves a new species.

Finally, the GA provides as much solutions as species remains actives over the search space.

FUTURE TRENDS

Since there are not any methods that provide the best results in all the possible situations, new approaches would be developed.

New fitness functions would help to locate a great number of valid solutions within the search space. In the described approaches this functions remains constants over the method execution. Another option would be allow dynamical fitness functions that vary along the execution stage. These kind of functions will try to adapt their output with the knowledge extracted from the search space while the crossover and mutation operators explore new arenas.

If different techniques offer acceptable solutions, other interesting approach an interesting point consists on putting together. For example, this hybrid models would integrate statistics methods (with a great mathematical background) with other heuristics.

CONCLUSION

This article shows an overview of the different methods related with evolutionary techniques used to address the problem of multimodality. This chapter showed several approaches to provide, not only a global solution, but multiple solutions to the same problem. It would help the final user to decide which of them is the most suitable in any particular case.

The final decision will depend on several factors, not only the global error reached for a particular method. Other factors also depend on the economic impact, the difficulty to implement it, the quality of the knowledge provided for their analysis, and so on.

REFERENCES

Ballester, P.J., & Carter, J.N. (2003). Real-Parameter Genetic algorithm for Finding Multiple Optimal Solutions in Multimodel Optimizaton, *Proceedings of Genetic and Evolutionary Computation*, pp. 706-717.

Bersano-Beguey, T. (1997) Controlling Exploration, Diversity and Escaping from Local Optimal in GP. *Proceedings of Genetic Prograrnming*. MIT Press. Cambridge, MA.

Deb, K., & Agrawal, S. (1995). Simulated binary crossover for continuous search space. *Complex Systems 9(2)*, pp. 115-148. 1995.

Deb, K., Anand, A., & Joshi, D. (2002). *A Computationally Efficient Evolutionary Algorithm for Real Parameter Optimization*, KanGAL report: 2002003.

DeJong, K.A. (1975). *An Analysis of the Behaviour of a Class of Genetic Adaptative Systems*. Phd. Thesis, University of Michigan, Ann Arbor.

Eshelman, L.J., & Schaffer J.D. (1994). Real coded genetic algorithms and interval schemata. *Foundations of Genetic Algorihtms (2)*, pp. 187-202.

Goldberg, D.E., & Richardson J. (1987) Genetic algorithms with Sharing for Multimodal Function Optimi-

zation. *Proceedings of 2nd International Conference on Genetic algorithms (ICGA)*, pp. 41-49.

Goldberg, D.E., & Wang, L. (1989). *Genetic algorithms in Search Optimization & Machine Learning*. Addison-Wesley.

Harik, G. (1995). Finding multimodal solutions using restricted tournament selection. *Proceedings of the Sixth International Conference on Genetic algorithms*, (ICGA) 24-31.

Landgon, W. (1996). *Evolution & Genetic Programming Populations*. University College. Technical Report RN/96/125. London.

Mengshoel, O.J., & Goldberg, D.E. (1999). Probabilistic Crowding: Deterministic Crowding with Probabilistic Replacement", Proceedings of Genetic and Evolutionary Computation, pp. 409-416.

Miller, B., & Shaw, M. (1995). *Genetic algorithms with Dynamic Niche Sharing for Multimodal Function Optimization*. IlliGAL Report 95010. University of Illinois. Urbana Champaign.

Ono, I., & Kobayashi, S. (1999). A real-coded genetic algorithm for function optimization using unimodal normal distribution. *Proceedings of International Conference on Genetic algorithms*, pp. 246-253.

Ortiz, D., Hervás, C., & García, N., (2005). CIXL2: A crossover operator for evolutionary algorithms based on population features. *Journal of Artificial Intelligence Research*.

Petrowski, A. (1996). A Clearing Procedure as a Niching Method for Genetic algorithms. *Proceedings of International Conference on Evolutionary Computation*. IEEE Press. Nagoya, Japan.

Ursem, R.K. (2002). Diversity-Guided Evolutionary Algorithms. *Proceedings of VII Parallel Problem Solving from Nature*, pp. 462-471.

KEY TERMS

Crossover: Genetic operation included in evolutionary techniques used to generate the offspring from current population. There are very different methods to perform crossover, but the general idea resides in merging the genetic information of the parents within the offspring with the aim of produce better solutions as generations advance.

Evolutionary Technique: Technique which tries to provide solutions for a problem guided by biological principles such as the survival of the fittest. This kind of techniques starts from a randomly generated population which evolves by means of crossover and mutation operations to provide the final solution.

Genetic Algorithm: A special type of evolutionary technique which represents the potential solutions of a problem within chromosomes (usually a collection of binary, natural or real values).

Multimodal Problems: A special kind of problems where a unique global solution does not exist. Several global optimums or one global optimum with several local optimums (or peaks) can be found around the search space.

Mutation: The other genetic operation included in evolutionary techniques to perform the reproduction stage. Mutation operator introduces new information in the system by random changes applied within the genetic individuals.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in. Combination of all the possible values for all the variables related with the problem.

Species: Within the context of genetic algorithm, a subset of genetic individuals with similar genotype (genetic values) which explore the same, or a similar, area of the search space.

Full–Text Search Engines for Databases

László Kovács
University of Miskolc, Hungary

Domonkos Tikk
Budapest University of Technology and Economics, Hungary

INTRODUCTION

Current databases are able to store several Tbytes of free-text documents. The main purpose of a database from the user's viewpoint is the efficient information retrieval. In the case of textual data, information retrieval mostly concerns the selection and the ranking of documents. The selection criteria can contain elements that apply to the content or the grammar of the language. In the traditional database management systems (DBMS), text manipulation is restricted to the usual string manipulation facilities, i.e. the exact matching of substrings. Although the new SQL1999 standard enables the usage of more powerful regular expressions, this traditional approach has some major drawbacks. The traditional string-level operations are very costly for large documents as they work without task-oriented index structures.

The required full-text management operations belong to text mining, an interdisciplinary field of natural language processing and data mining. As the traditional DBMS engine is inefficient for these operations, database management systems are usually extended with a special full-text search (FTS) engine module. We present here the particular solution of Oracle; there for making the full-text querying more efficient, a special engine was developed that performs the preparation of full-text queries and provides a set of language and semantic specific query operators.

BACKGROUND

Traditional DBMS engines are not adequate to meet the users' requirements on the management of free-text data as they handles the whole text field as an atom (Codd, 1985). A special extension to the DBMS engine is needed for the efficient implementation of text manipulating operations. There is a significant demand on the market on the usage of free text and text mining operations, since information is often stored as free text. Typical application areas are, e.g., text analysis in medical systems, analysis of customer feedbacks, and bibliographic databases. In these cases, a simple character-level string matching would retrieve only a fraction of related documents, thus an FST engine is required that can identify the semantic similarities between terms.

There are several alternatives for implementing an FTS engine. In some DBMS products, such as Oracle, Microsoft SQLServer, Postgres, and mySQL, a built-in FTS engine module is implemented. Some other DBMS vendors extended the DBMS configuration with a DBMS-independent FTS engine. In this segment the main vendors are: SPSS LexiQuest (SPSS, 2007), SAS Text Miner (SAS, 2007), dtSearch (dtSearch, 2007), and Statistica Text Miner (Statsoft, 2007).

The market of FTS engines is very promising since the amount of textual information stored in databases rises steadily. According to the study of Meryll Lynch (Blumberg & Arte, 2003), 85% of business information are text documents – e-mails, business and research reports, memos, presentations, advertisements, news, etc. – and their proportion still increases. In 2006, there were more than 20 billion documents available on the Internet (Chang, 2006). The estimated size of the pool increases to 550 billion documents when the documents of the hidden (or deep) web – which are e.g. dynamically generated ones – are also considered.

TEXT MINING

The subfield of document management that aims at processing, searching, and analyzing text documents is *text mining*. The goal of text mining is to discover the non-trivial or hidden characteristics of individual documents or document collections. Text mining is an

Figure 1. The text mining module
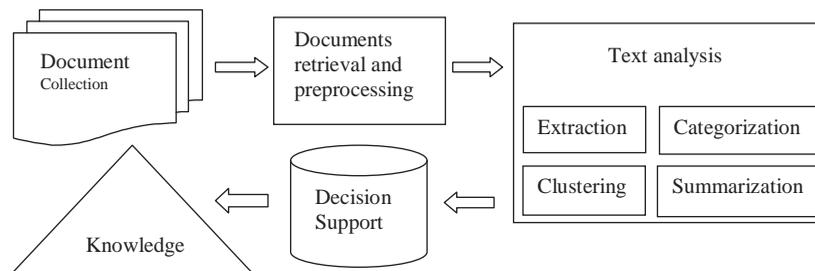
application oriented interdisciplinary field of machine learning which exploits tools and resources from computational linguistics, natural language processing, information retrieval, and data mining.

The general application schema of text mining is depicted in Figure 1 (Fan, Wallace, Rich & Zhang, 2006). For giving a brief summary of text mining, four main areas are presented here: information extraction, text categorization/classification, document clustering, and summarization.

Information Extraction

The goal of information extraction (IE) is to collect the text fragments (facts, places, people, etc.) from documents relevant to the given application. The extracted information can be stored in structured databases. IE is typically applied in such processes where statistics, analyses, summaries, etc. should be retrieved from texts. IE includes the following subtasks:

- named entity recognition – recognition of specified types of entities in free text, see e.g. Borthwick, 1999; Sibanda & Uzuner, 2006,
- co-reference resolution – identification of text fragments referring to the same entity, see e.g. Ponzetto & Strube, 2006,
- identification of roles and their relations – determination of roles defined in event templates, see e.g. Ruppenhofer et al, 2006.

Text Categorization

Text categorization (TC) techniques aim at sorting documents into a given category system (see Sebastiani, 2002 for a good survey). In TC, usually, a classifier

model is built based on the content of a set of sample documents, which model is then used to classify unseen documents. Typical application examples of TC include among many others:

- document filtering – such as e.g. spam filtering, or newsfeed (Lewis, 1995);
- patent document routing – determination of experts in the given fields (Larkey, 1999);
- assisted categorization – helping domain experts in manual categorization with valuable suggestions (Tikk et al, 2007),
- automatic metadata generation (Liddy et al, 2002),

Document Clustering

Document clustering (DC) methods group elements of a document collection based on their similarity. Here again, documents are usually clustered based on their content. Depending on the nature of the results, one can have partitioning and hierarchical clustering methods. In the former case, there is no explicit relation among the clusters, while in the latter case a hierarchy of clusters is created. DC is applied for e.g.:

- clustering the results of (internet) search for helping users in locating information (Zamir et al, 1997),
- improving the speed of vector space based information retrieval (Manning et al, 2007),
- providing a navigation tool when browsing a document collection (Käki, 2005).

Summarization

Text summarization aims at the automatic generation of short and comprehensible summaries of documents. Text extraction algorithms create summary by extracting relevant descriptive phrases (typically sentences) from the original text, while summaries generated by abstraction methods may contain synthesized text as well. The typical application areas of summarization span from the internet search to arbitrary document management system (Ganapathiraju, 2002; Radev et al; 2001).

FULL-TEXT SEARCH (FTS) ENGINES

Full-Text Search

Based on the literature (Maier, 2001, Curtmola, 2005), an effective FTS engine should support several query functionalities. The simplest operation is the string-based query, which retrieves texts that exactly match the query string. In some cases, the position of the keywords within the document is also an important factor. The simplest form of similarity-based matching uses the edit-distance function. The next operation is the content-based query, where similarity is defined on the semantic level. An FTS engine should also support grammar (and therefore language) specific operators (e.g. stemming). The highest level of text search operates with semantic-based matching (thesaurus-based neighborhood, generalization of a word, specialization, synonyms). From the practical viewpoint, the efficient execution of queries is also very important. Due to the heterogeneity of the source pool, the support of different document formats is a key requirement. The minimal usage of other resources provides an independent, flexible solution. From the aspect of software development, the open, standardized interface is a good investment. To provide a manageable, easy to understand response, the efficient ranking of the result set is crucial (Chakrabarti, 2006). The products and test systems currently available only partially meet the above requirements.

Structure of a General FTS Engine

FTS engines are structurally similar to database systems: they store data and metadata; their purpose is to provide an efficient information retrieval (Microsoft, 2007; Oracle Text, 2007). As the processing of a full-text query requires several distinct steps, the FTS engines typically have modular structure (see also Figure 2.).

The *loader* module loads the documents into a common staging area, into a common representation. In further steps, data items are transformed into a common format, too. The loaded documents are stored in the *datastore unit*. Document processing has several steps. The *sectioner unit* has to discover the larger internal logical structure of the documents. The *word-breaker* parses the text into smaller syntactical units like paragraphs, sentences and terms (words). For reducing the length and complexity of the text, several preprocessing steps are executed. First, a *filter module* is applied that discard irrelevant words (stop-words, noise words). Next, the *stemmer unit* generates the stem form for every word. In the background, the *language lexicon* supports the language-specific reduction steps. This lexicon contains the grammar of the supported languages and the list of stop-words. The thesaurus is a special lexicon, which stores the terms organized in a graph based on their semantic relationship. To provide an efficient term management, several kinds of indexes are created. The *indexer unit* manages the different document-term indices that enable the efficient access to term occurrences. On the front-end side, the *query preprocessor* transforms the user's query into an internal format. This format is processed by the *query matcher*, resulting in a set of matching documents. The search engine may be extended with a text mining module that performs data mining operations, like clustering or classification. In order to provide a more accurate response, the *query refinement engine* performs the processing of relevance feedback. The list of matching documents is pipelined to the *ranking module*. The *exporter module* generates the final format of the ranked document set.

As mentioned, database systems use indices for the fast access to data items. For full-text search, the inverted index is the most efficient index structure (Zobel, 2006). In the simple inverted index, the key of the index is the term. Each key is associated with a pair (df, dl). Here df is the number of documents containing the key, and dl is the list of documents that contain the key. Each entry in the list contains a document identifier and the frequency value in the document. The position-based inverted index differs from the simple version as that the list corresponding to a document also contains the positions of the given term in the text.

Figure 2. Modules of an FTS engine
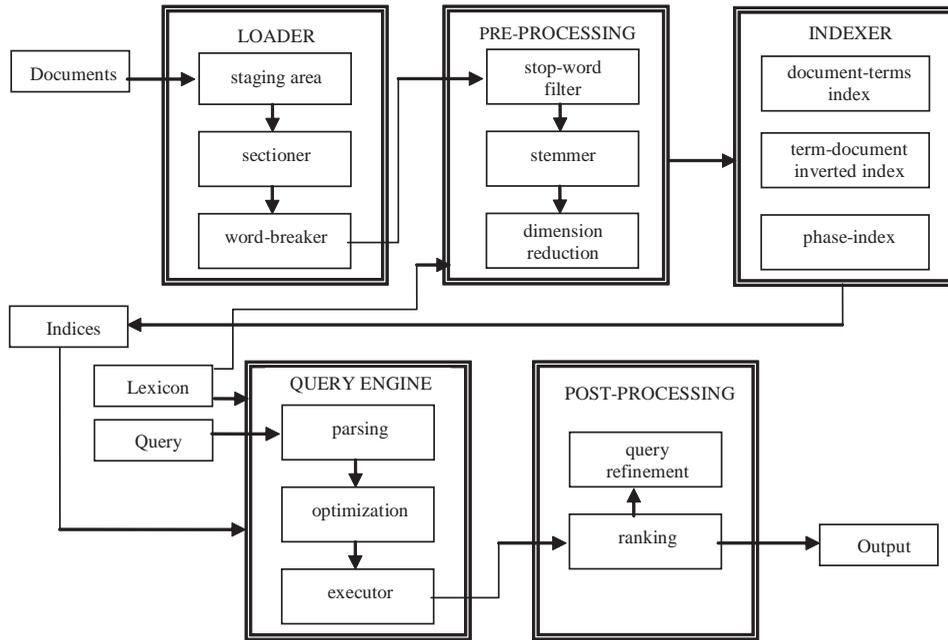
FTS Engine Interface in Oracle Text

The FTS functionality in Oracle Text (Oracle, 2007) can be activated with some extensions to SQL and with procedural SQL packages. Oracle Text supports four index types:

- CONTEXT-type index: inverted index for long documents;
- CTXCAT-type index: to support content- and attribute-based indexing for shorter documents;
- CTXRULE-type index: rules for document clustering;
- CTXPATH-type index: indexing of XML documents.

The stemming module supports only two languages: English and French. In the queries, the CONTAINS operator supports the following matching modes:

- keyword: exact matching;
- AND, OR, NOT : Boolean operators;
- NEAR (keyword1, keyword2): the keywords should occur at near positions in the document;
- BT(keyword): generalization of the keyword;

- NT(keyword): specialization of the keyword;
- REL(keyword): words in the thesaurus in relation with the keyword;
- SYN(keyword): the synonyms of the keyword;
- \$keyword: words having the same stem;
- !keyword: words having the same pronunciation;
- ABOUT keywords: words belonging to the given topic;
- FUZZY(keyword): words that are close to the keyword in terms of the edit distance;
- WITHIN (section): the matching is restricted to a given section of the documents.

The example below retrieves the documents containing words that have similar meaning as "food":

SELECT description FROM books WHERE CONTAINS (description, 'NT(food,1)') > 0;

Oracle Text supports three methods for document partition (categorization & clustering). The manual categorization allows the user to enter keyword-category pairs. The automatic categorization works if a training set of document-category pairs is given. The cluster-

ing method automatically determines the clusters in a set of documents based on their similarity. To provide semantic-based matching for any arbitrary domain, the users can create their own thesaurus.

FUTURE TRENDS

In our view, there are three main areas where the role of FTS engine should be improved in the future: web search engines, ontology-based information retrieval, and management of XML documents. The main standard for the query of XML documents is nowadays the XQuery language. This standard is very flexible for selecting structured data elements, but it has no special features for the unstructured part. In (Botev, 2004; Curtmola, 2005), an extension of XQuery with full-text functionality is proposed. The extended query language is called TeXQuery and GalaTex. The language contains a rich set of composite full-text primitives such as phrase matching, proximity distance, stemming and thesauri. The combination of structure- and content-based queries is investigated deeply from a theoretical viewpoint in (Amer, 2004).

The efficiency of information retrieval can be improved with the extension of additional semantic information. The ALVIS project (Luu, 2006) aims at building a distributed, peer-to-peer semantic search engine. The peer-to-peer network is a self-organizing system for decentralized data management in distributed environments. During a query operation, a peer broadcasts search requests in the network. A peer may be assigned to a subset of data items. The key element in the cost reduction is the application of a special index type at the nodes. The index contains in addition to the single keyword entries also entities for compound keys with high discriminative values.

A very important application area of full-text search is the Web. A special feature of Web search is that the users apply mostly simple queries. Only 10% of queries use some complex full-text primitives like Boolean operators, stemming or fuzzy matching. Eastman (2003) investigated the reasons of omitting the complex operators and concluded that the application of complex full-text operators does not significantly improve the search results. Efficiency is a key factor in web search engines (Silvestri, 2004). The goal of the research is to upgrade the indexing mechanism of web search engines to provide efficient full-text search operators.

CONCLUSION

The information is stored on the web and in computers mostly in free-text format. The current databases are able to store and manage huge document collection. Free-text data sources require specific search operations. Database management systems usually contain a separate full-text search engine to perform full-text search primitives. In general, the current FTS engines support the following functionalities: exact matching, position-based matching, similarity-based matching (fuzzy matching), grammar-based matching (stemming) and semantic-based matching (synonym- and thesaurus-based matching). It has been shown that the average user requires additional help to exploit the benefits of these extra operators. Current research focuses on solving the problem of covering new document formats, adapting the query to the user's behavior, and providing an efficient FTS engine implementation.

REFERENCES

Amer Yahia, S., Lakshmanan, L. & Pandit, S. (2004). FlexPath: Flexible Structure and Full-Text Querying for XML. In *Proc. of ACM SIGMOD* (pp.83–94), Paris, France.

Borthwick, A. (1999). A Maximum Entropy Approach to Named Entity Recognition, Ph.D. thesis. New York University, USA.

Blumberg, R. & Arte, S. (2003). The problem with unstructured data. *DM Review* (February).

Botev, C., Amer-Yaiha, S. & Shanmugasundaram, J. (2004). A TexQuery-based XML full-text search engine. In *Proc. of ACM SIGMOD* (pp. 943–944), Paris, France.

Chakrabarti, K., Ganti, V., Han, J. & Xin, D. (2006). Ranking Objects by Exploiting Relationships: Computing Top-K-over Aggregation. In *Proc. of ACM SIGMOD* (pp. 371–382), Chicago, IL, USA.

Chang, K. & Cho, J. (2006). Accessing the Web: From Search to Integration, In *Proc. of ACM SIGMOD* (pp. 804–805) , Chicago, IL, USA.

Codd, E.F (1985). Is Your DBMS Really Rational, (Codd's 12 rules), *Computerworld Magazine*

Curtmola, E., Amer-Yaiha, S., Brown, P. & Fernandez, M. (2005). GalaTex: A Conformant Implementation of the Xquery Full-Text Language, *Proc. of WWW 2005.*(pp. 1024–1025), Chiba, Japan.

dtSearch (2007), Text Retrieval / Full Text Search Engine, http://www.dtsearch.com

Eastman, C. & Jansen, B (2003). Coverage, Relevance, and Ranking: The Impact of Query Operators on Web Search Engine Results, *ACM Transactions on Information Systems*, *21*, (4), 383–411.

Fan, W., Wallace, L., Rich, S., & Zhang, Z. (2006). Tapping the power of text mining. *Communications of the ACM*, *49* (9), 76–82.

M. K. Ganapathiraju (2002). Relevance of cluster size in MMR based summarizer. Technical Report 11-742, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA.

Käki, M. (2005). Findex: search result categories help users when document ranking fails. In *CHI-05: Proc. of the SIGCHI conference on Human factors in computing systems* (pp. 131–140), Portland, OR, USA.

Larkey, L. S. (1999). A patent search and classification system. In *Proc. of DL-99, 4th ACM Conference on Digital Libraries* (pp. 179–187), Berkeley, CA, USA.

Lewis, D. D. (1995). The TREC-4 filtering track: description and analysis. In *Proc. of TREC-4, 4th Text Retrieval Conference,* (pp. 165–180), Gaithersburg, MD, USA.

Liddy, E.D., Sutton, S., Allen, E., Harwell, S., Corieri, S., Yilmazel, O., Ozgencil, N.E., Diekema, A., McCracken, N., & Silverstein, J. (2002). Automatic metadata generation and evaluation. In *Proc. of ACM SIGIR* (pp. 401–402), Tampere, Finland.

Luu, T., Klemm, F., Podnar, I., Rajman, M. & Aberer, K. (2006). ALVIS Peers: A Scalable Full-text Peer-to-Peer Retrieval Engine, *Proc. of ACM P2PIR'06* (pp. 41–48), Arlington, VA, USA.

Maier, A.; Simmen, D. (2001). DB2 Optimization in Support of Full Text Search, Bulletin of IEEE on Data Engineering.

Manning, Ch. D., Raghavan, P., & Schütze, H. (2007). Introduction to Information Retrieval. Cambridge University Press.

Microsoft (2007). SQL Server Full Text Search Engine, http://technet.microsoft.com/en-us/library/ms345119.aspx

Oracle Text (2007). *Oracle Text Product Description*, homepage: http://www.oracle.com/technology/products/text/index.html

Ponzetto, S. P., & Strube, M. (2006). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proc. of HLT-NAACL, Human Language Technology Conf. of the NAACL* (pp. 192–199), New York, USA.

Radev, D., Blair-Goldensohn, S., & Zhang, Z. (2001). Experiments in single and multi-document summarization using MEAD. In *Proc. of DUC-01, Document Understanding Conf., Workshop on Text Summarization*, New Orleans, USA.

Ruppenhofer, J., Ellsworth, M., Petruck, M. R. L., Johnson, Ch. R., & Scheffczyk, J. (2006). FrameNet II: Extended Theory and Practice. International Computer Science Institute, Berkeley, USA.

SAS (2007). SAS Text Miner, http://www.sas.com/technologies/analytics/datamining/textminer/

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47.

Sibanda, T., & Uzuner, Ö. (2006). Role of local context in automatic deidentification of ungrammatical, fragmented text. In *Proc. of HLT-NAACL, Human Language Technology Conf. of the NAACL* (pp. 65–73), New York, USA.

Silvestri, F., Orlando, S & Perego, R. (2004). WINGS: A Parallel Indexer for Web Contents, *Lecture Notes in Computer Science, 3036*, pp. 263-270.

SPSS (2007). Predictive Text Analysis, http://www.spss.com/predictive_text_analytics/

Statsoft (2007). STATISTICA Text Miner, http://www.statsoft.com

Tikk, D., Biró, Gy., & Törcsvári, A. (2007). A hierarchical online classifier for patent categorization. In do

Prado, H. A. & Ferneda, E., editors, *Emerging Technologies of Text Mining: Techniques and Applications*. Idea Group Inc. (in press).

Zamir, O., Etzioni, O., Madani, O., & Karp, R. M. (1997). Fast and intuitive clustering of web documents. In *Proc. of SIGKDD-97, 3rd Int. Conf. on Knowledge Discovery and Data Mining* (pp. 287–290), Newport Beach, USA.

Zobel, J. & Moffat, A. (2006). Inverted Files for Text Search Engines, *ACM Computing Surveys*, *38*(2), Article 6.

KEY TERMS

Full-Text Search (FTS) Engine: A module within a database management system that supports efficient search in free texts. The main operations supported by the FTS engine are the exact matching, position-based matching, similarity-based matching, grammar-based matching and semantic-based matching.

Fuzzy Matching: A special type of matching where the similarity of two terms are calculated as the cost of the transformation from one into the other. The most widely used cost calculation method is the edit distance method.

Indexer: It builds one or more indices for the speed up information retrieval from free text. These indices usually contain the following information: terms (words), occurrence of the terms, format attributes.

Inverted Index: An index structure where every key value (term) is associated with a list of objects identifiers (representing documents). The list contains objects that include the given key value.

Query Refinement Engine: A component of the FTS engine that generates new refined queries to the initial query in order to improve the efficiency of the retrieval. The refined queries can be generated using the users' response or some typical patterns in the query history.

Ranking Engine: A module within the FTS engine that ranks the documents of the result set based on their relevance to the query.

Sectioner: A component of the FTS engine, which breaks the text into larger units called sections. The types of extracted sections are usually determined by the document type.

Stemmer: It is a language-dependent module that determines the stem form of a given word. The stem form is usually identical to the morphological root. It requires a language dictionary.

Thesaurus: A special repository of terms, which contains not only the words themselves but the similarity, the generalization and specialization relationships. It describes the context of a word but it does not give an explicit definition for the word.

Word-Braker: A component of the full-text engine whose function is to break the text into words and phrases.

Functional Dimension Reduction for Chemometrics

Tuomas Kärnä
Helsinki University of Technology, Finland

Amaury Lendasse
Helsinki University of Technology, Finland

INTRODUCTION

High dimensional data are becoming more and more common in data analysis. This is especially true in fields that are related to spectrometric data, such as chemometrics. Due to development of more accurate spectrometers one can obtain spectra of thousands of data points. Such a high dimensional data are problematic in machine learning due to increased computational time and the curse of dimensionality (Haykin, 1999; Verleysen & François, 2005; Bengio, Delalleau, & Le Roux, 2006).

It is therefore advisable to reduce the dimensionality of the data. In the case of chemometrics, the spectra are usually rather smooth and low on noise, so function fitting is a convenient tool for *dimensionality reduction*. The fitting is obtained by fixing a set of basis functions and computing the fitting weights according to the least squares error criterion.

This article describes a unsupervised method for finding a good function basis that is specifically built to suit the data set at hand. The basis consists of a set of Gaussian functions that are optimized for an accurate fitting. The obtained weights are further scaled using a Delta Test (DT) to improve the prediction performance. Least Squares Support Vector Machine (LS-SVM) model is used for estimation.

BACKGROUND

The approach where multivariate data are treated as functions instead of traditional discrete vectors is called Functional Data Analysis (FDA) (Ramsay & Silverman, 1997). A crucial part of FDA is the choice of basis functions which allows the functional representation. Commonly used bases are B-splines (Alsberg & Kvalheim, 1993), Fourier series or wavelets (Shao, Leung, & Chau, 2003). However, it is appealing to build a problem-specific basis that employs the statistical properties of the data at hand.

In literature, there are examples of finding the optimal set of basis functions that minimize the fitting error, such as Functional Principal Component Analysis (Ramsay et al., 1997). The basis functions obtained by Functional PCA usually have global support (i.e. they are non-zero throughout the data interval). Thus these functions are not good for encoding spatial information of the data. The spatial information, however, may play a major role in many fields, such as spectroscopy. For example, often the measured spectra contain spikes at certain wavelengths that correspond to certain substances in the sample. Therefore these areas are bound to be relevant for estimating the quantity of these substances.

We propose that locally supported functions, such as Gaussian functions, can be used to encode this sort of spatial information. In addition, variable selection can be used to select the relevant functions from the irrelevant ones. Selecting important variables directly on the raw data is often difficult due to high dimensionality of data; computational cost of variable selection methods, such as Forward-Backward Selection (Benoudjit, Cools, Meurens, & Verleysen, 2004; Rossi, Lendasse, François, Wertz, & Verleysen, 2006), grows exponentially with the number of variables. Therefore, wisely placed Gaussian functions are proposed as a tool for encoding spatial information while reducing data dimensionality so that other more powerful information processing tools become feasible. Delta Test (DT) (Jones, 2004) based scaling of variables is suggested for improving the prediction performance.

A typical problem in chemometrics deals with predicting some chemical quantity directly from measured spectrum. Due to additivity of absorption spectra, the problem is assumed to be linear and therefore linear

models, such as Partial Least Squares (Härdle, Liang, & Gao, 2000) have been widely used for the prediction task. However, it has been shown that the additivity assumption is not always true and environmental conditions may further introduce more non-linearity to the data (Wülfert, Kok, & Smilde, 1998). We therefore propose that in order to address a general prediction problem, a non-linear method should be used. LS-SVM is a relatively fast and reliable non-linear model which has been applied to chemometrics as well (Chauchard, Cogdill, Roussel, Roger, & Bellon-Maurel, 2004).

USING GAUSSIAN BASIS WITH SPECTOMETRIC DATA

Consider a problem where the goal is to estimate a certain quantity $p \in \Re$ from a measured absorption spectrum X based on the set of N training examples $(X_j, p_j)_{j=1}^N$. In practice, the **spectrometric data** X_j is a set of discretized measurements $(x_i^j, y_i^j)_{i=1}^m$ where $x_i^j \in [a,b] \subset \Re$ stand for the observation wavelength and $y_i^j \in \Re$ is the response.

Adopting the FDA framework (Ramsay et al., 1997), our goal is to build a prediction model F so that $\hat{p} = F(X)$. Here, the argument X is a real-world spectrum, i.e. a continuous function that maps wavelengths to responses. Without much loss of generality it can be assumed that X belongs to $L_2([a, b])$, the space of square integrable functions on the interval [a,b]. However, since the spectrum X is unknown and infinite dimensional it is impossible to build the model F(X) in practice. Therefore X must be approximated with a q dimensional representation $\omega = P(X), P : L_2 \rightarrow \Re^q$, and our prediction model becomes $\hat{p} = F(\omega)$. Naturally, in order to obtain **dimensionality reduction**, we

require that q is smaller than the number of points in the spectra.

Figure 1 presents a graph of the overall prediction method. Gaussian fitting is used for the approximation of X. The obtained vectors ω are further scaled by a diagonal matrix A before the final LS-SVM modeling. The following sections explain these steps in greater detail.

Gaussian Fitting: Approximating Spectral Function X

Because the space $L_2([a, b])$ is infinite dimensional function space, it is necessary to consider some finite dimensional subspace $V \subset L_2([a, b])$ in order to obtain a feasible **function approximation**. We define V by a set of **Gaussian functions**

$$\varphi_k(x) = e^{-\|x - t_k\|^2 / \sigma_k^2}, k = 1, \ldots, q, \qquad (1)$$

where t_k is the center and σ_k is the width parameter. The set $\varphi_k(x)$ spans a q dimensional normed vector space and we can write $V = \text{span}\{\varphi_k(x)\}$. A natural choice for the norm is the L_2 norm:

$$\left\|\hat{f}\right\|_V = (\int_a^b \hat{f}(x)^2 dx)^{1/2}.$$

Now X can be approximated using the basis representation $\hat{X}(x) = \omega^T \phi(x)$, where

$$\phi(x) = [\varphi_1(x), \varphi_2(x), \ldots, \varphi_q(x)]^T.$$

The weights ω are chosen to minimize the square error:

Figure 1. Outline of the prediction method
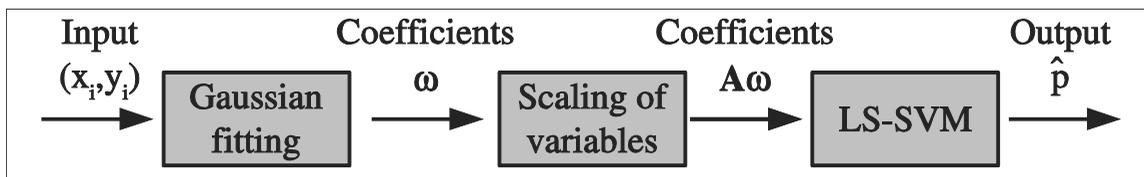
$$\min_{\omega} \sum_{i=1}^{m} \left| y_i - \omega^T \phi(x_i) \right|^2 . \tag{2}$$

In other words, we simply fit a function to the points $(x_i, y_i)_{i=1}^m$ using the **basis functions** $\varphi_k(x)$. Now, any function $\hat{X} \in V$ is uniquely determined by the weight vector ω. This suggests that it is equivalent to analyze the discrete weight vectors ω instead of the continuous functions $\hat{X}$.

Orthonormalization

Radial symmetric models (such as the LS-SVM) depend only on the distance metric $d(\cdot,\cdot)$ in the input space. Thus, we require that the mapping from V to $\Re^q$ is isometric, i.e. $d_V(\hat{f}, \hat{g}) = d_q(\alpha, \beta)$ for any functions $\hat{f}(x) = \alpha^T \phi(x)$ and $\hat{g}(x) = \beta^T \phi(x)$. The first distance is calculated in the function space and the latter one in $\Re^q$. In the space V, distances are defined by the norm $d(\hat{f}, \hat{g}) = \left\| \hat{f} - \hat{g} \right\|_V$. Now a simple calculation gives

$$\left\| \hat{f} - \hat{g} \right\|_V^2 =$$

$$\int_a^b \left(\sum_{k=1}^{q} (\alpha - \beta) \varphi_k(x) \right)^2 dx = (\alpha - \beta)^T \Phi (\alpha - \beta) ,$$

where

$$\Phi_{i,j} = \int_a^b \varphi_i(x) \varphi_j(x) dx .$$

This implies that if the **basis** is **orthonormal**, the matrix Φ becomes an identity matrix and the distances become equal, i.e.

$$\left\| \hat{f} - \hat{g} \right\|_V = \left\| \alpha - \beta \right\|_q = ((\alpha - \beta)^T (\alpha - \beta))^{1/2} .$$

Unfortunately this is not the case with Gaussian basis and a linear transformation $\tilde{\omega} = U\omega$ need to be applied. Here the matrix U is the Cholesky decomposition of $\Phi = U^T U$. In fact, the transformed weights ω are related to a set of new basis functions $\tilde{\phi} = U^{-1}\phi$ that are both optimized to fit the data and orthonormal.

Finding an Optimal Gaussian Basis

When the **basis functions** are fixed, the weights ω are obtained easily by solving the problem (2). The solution is the pseudoinverse $\omega = (G^T G)^{-1} G^T y$ (Haykin, 1999), where $y = [y_1, y_2,..., y_m]^T$ are the values to be fitted and $[G]_{i,j} = \varphi_j(x_i)$.

Since the **Gaussian functions** are differentiable, the locations and widths can be optimized for a better fit. The average fitting error of all functions is obtained by averaging Eq. (2) over all of the sample inputs $j = 1, . . .,N$. Using the matrix notation given above, it can be formulated as

$$E = \frac{1}{2N} \sum_{j=1}^{N} (G\omega_j - y_j)^T (G\omega_j - y_j) ,$$

which can be differentiated with respect to t_k and σ_k (Kärnä & Lendasse, 2007).

Knowing the partial derivates, the locations and the widths can be optimized using unconstrained **nonlinear optimization**. In this article, **Broyden-Fletcher-Goldfarb-Shanno (BFGS)** Quasi-Newton method with line search is suggested. The formulation of the BFGS algorithm can be found in Bazaraa, Sherali and Shetty (1993).

An example of spectral data and an optimized basis functions in presented in Figure 2. This application is

Figure 2. Above: NIR absorption spectra. Below: 13 optimized basis functions
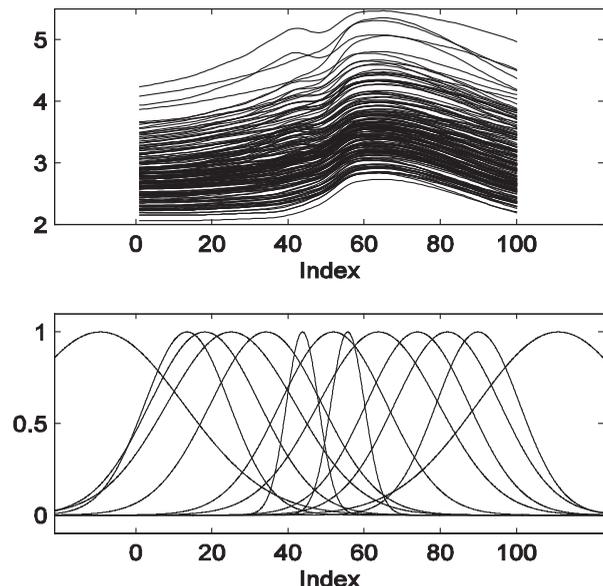
related to prediction of fat content in meat samples using NIR absorption spectra (Kärnä et al., 2007; Rossi et al., 2006; Thodberg, 1996). It can be seen that the basis has adapted to the data: there are narrow functions in the center where there is more variance in the data.

Variable Scaling

Variable scaling can be seen as a generalization of variable selection; in variable selection variables are either included in the training set (corresponding to multiplication by 1) or excluded from it (corresponding to multiplication by 0), while in variable scaling the entire range [0,1] of scalars is allowed. In this article, we present a method for choosing the scaling using Delta Test (DT) (Lendasse, Corona, Hao, Reyhani, & Verleysen, 2006).

The scalars are generated by iterative ***Forward-Backward Selection*** (FBS) (Benoudjit et al., 2004; Rossi et al., 2006). FBS is usually used for variable selection, but it can be extended to scaling as well; Instead of turning scalars from 0 to 1 or vice versa, increases by $1/h$ (in the case of forward selection) or decreases by $1/h$ (in the case of backward selection) are allowed. Integer h is a constant grid parameter. Starting from an initial scaling, the FBS algorithm changes the each of the scalars by $\pm 1/h$ and accepts the change that resulted in the best improvement. The process in repeated until no improvement is found. The process is initialized with several sets of random scalars.

DT is a method for estimating the variance of the noise within a data set. Having a set of general input-output pairs $(\mathbf{x}_i, y_i)_{i=1}^N \in \Re^m \times \Re$ and denoting the nearest neighbor of $\mathbf{x}_i$ by $\mathbf{x}_{NN(i)}$, the DT variance estimate is

$$\delta = \frac{1}{2N} \sum_{i=1}^N \left| y_{NN(i)} - y_i \right|^2,$$

where $y_{NN(i)}$ is the output of $\mathbf{x}_{NN(i)}$. Thus, δ is equivalent to the residual (i.e. prediction error) of a first-nearest-neighbor model. DT is useful in evaluation of dependence of random variables and therefore it can be used for scaling: The set of scalars that give the smallest δ is selected.

LS-SVM

LS-SVM is a least square modification of the Support Vector Machine (SVM) (Suykens, Van Gestel, De Brabanter, De Moor, & Vandewalle, 2002). The quadratic optimization problem of SVM is simplified so that it reduces into a linear set of equations. Moreover, regression SVM usually involves three unknown parameters while LS-SVM has only two; the regularization parameter γ and the width parameter θ.

Given a set of N training examples $(\mathbf{x}_i, y_i)_{i=1}^N \in \Re^m \times \Re$ the LS-SVM model is $\hat{y} = \mathbf{w}^T \psi(x) + b$, where $\psi : \Re^m \to \Re^n$ is a mapping from the input space onto a higher dimensional hidden space, $\mathbf{w} \in \Re^n$ is a weight vector and b is a bias term. The optimization problem is formulated as

$$\underset{\mathbf{w},b}{\text{Min}} \; J(\mathbf{w},b) = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{2}\gamma \sum_{i=1}^N e_i^2$$

$$\text{so that } y_i = \mathbf{w}^T \psi(x_i) + b + e_i,$$

where e_i is the prediction error and $\gamma \geq 0$ is a regularization parameter. The dual problem is derived using Lagrangian multipliers which lead into a linear KKT system that is easy to solve (Suykens et al., 2002). Using the dual solution, the original model can be reformatted as

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^N \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b,$$

where the kernel $K(\mathbf{x}, \mathbf{x}_i) = \psi(\mathbf{x})^T \psi(\mathbf{x}_i)$ is a continuous and symmetric mapping from $\Re^m \times \Re^m$ to $\Re$ and α_i are the Lagrange multipliers. A widely-used choice for the K is the standard Gaussian kernel $K(\mathbf{x}_1, \mathbf{x}_2) = e^{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 / \theta^2}$.

The LS-SVM prediction is the final step in the proposed method where spectral data is compressed by the Gaussian fitting and the fitting weights are normalized and scaled before the prediction. More elaborate discussion and applications to real-world data are presented in Kärnä et al. (2007).

FUTURE TRENDS

The only unknown parameter in the proposed method is the number of basis functions which is selected by validation. In future other methods for determining good basis size should be developed in order to speed up the process. Moreover, the methodology should be tested with various data sets, including other than

F

spectral data. The LS-SVM predictor could be also replaced with another model.

Although the proposed Gaussian fitting combined with LS-SVM model seems to be fairly robust, the relation between the basis functions and the prediction performance should be studied in detail. It would be desirable to optimize the basis directly for best possible prediction performance (instead of good data fitting), although it seems difficult due to over-fitting and high computational costs.

CONCLUSION

This article deals with the problem of finding a good set of basis functions for dimension reduction of spectral data. We have proposed a method based on Gaussian basis functions where the locations and the widths of the functions are optimized to fit the data as accurately as possible. The basis indeed tends to follow the nature of the data and provides a good tool for dimension reduction. Other methods, such as the proposed DT scaling, will benefit from the smaller data dimension and help to achieve even better data compression. The LS-SVM model is a robust and fast method to be used in the final prediction.

REFERENCES

Alsberg, B. K., & Kvalheim, O. M. (1993). Compression of nth-order data arrays by B-splines. I : Theory. *Journal of Chemometrics* 7, 61–73.

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (1993). *Nonlinear Programming, Theory and Algorithms*. John Wiley and Sons.

Bengio, Y., Delalleau, O., & Le Roux, N. (2006). The Curse of Highly Variable Functions for Local Kernel Machines. Y. Weiss and B. Schölkopf and J. Platt (editors), *Neural Information Processing Systems (NIPS 2005)*, Advances in Neural Information Processing Systems 18, 107–114.

Benoudjit, N., Cools, E., Meurens, M., & Verleysen, M. (2004). Chemometric calibration of infrared spectrometers: selection and validation of variables by non-linear models. *Chemometrics and Intelligent Laboratory Systems* 70, 47–53.

Chauchard, F., Cogdill, R., Roussel, S., Roger, J. M., & Bellon-Maurel, V. (2004). Application of LS-SVM to non-linear phenomena in NIR spectroscopy: development of a robust and portable sensor for acidity prediction in grapes. *Chemometrics and Intelligent Laboratory Systems* 71, 141–150.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall.

Härdle, W., Liang, H., & Gao, J. T. (2000). *Partially Linear Models*. Physica-Verlag.

Jones, A. J. (2004). New tools in non-linear modeling and prediction. *Computational Management Science* 1, 109–149.

Kärnä, T., & Lendasse, A. (2007). Gaussian Fitting Based FDA for Chemometrics. F. Sandoval, A. Prieto, J. Cabestany, M. Graña (editors), *9th International Work-Conference on Artificial Neural Networks (IWANN'2007)*, Lecture Notes in Computer Science 4507, 186–193.

Lendasse, A., Corona, F., Hao, J., Reyhani, N., & Verleysen, M. (2006). Determination of the Mahalanobis matrix using nonparametric noise estimations. M. Verleysen (editor), *14th European Symposium on Artificial Neural Networks (ESANN 2006),* d-side publi., 227–232.

Ramsay, J., & Silverman, B. (1997). *Functional Data Analysis*. Springer Series in Statistics. Springer.

Rossi, F., Lendasse, A., François, D., Wertz, V., & Verleysen, M. (2006). Mutual information for the selection of relevant variables in spectrometric nonlinear modeling. *Chemometrics and Intelligent Laboratory Systems* 80 (2), 215–226.

Shao, X. G., Leung, A. K., & Chau, F. T. (2003). Wavelet: A New Trend in Chemistry. *Accounts of Chemical Research* 36, 276–283.

Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least Squares Support Vector Machines*. World Scientific Publishing.

Thodberg, H. (1996). A Review of Bayesian Neural Networks with an Application to Near Infrared Spectroscopy. *IEEE Transactions on Neural Networks* 7, 56–72.

Verleysen, M., & François, D. (2005). The Curse of Dimensionality in Data Mining and Time Series Prediction. J. Cabestany, A. Prieto, and D.F. Sandoval (editors), *8th International Work-Conference on Artificial Neural Networks (IWANN'2005)*, Lecture Notes in Computer Science 3512, 758–770.

Wülfert, F., Kok, W. T., & Smilde, A. K. (1998). Influence of temperature on vibrational spectra and consequences for the predictive ability of multivariate models. *Analytical Chemistry* 70, 1761–1767.

KEY TERMS

Chemometrics: Application of mathematical or statistical methods to chemical data. Closely related to monitoring of chemical processes and instrument design.

Curse of Dimensionality: A theoretical result in machine learning that states that the lower bound of error that an adaptive machine can achieve increases with data dimension. Thus performance will degrade as data dimension grows.

Delta Test: A Non-parametric Noise Estimation method. Estimates the amount of noise within a data set, i.e. the amount of information that cannot be explained by any model. Therefore Delta Test can be used to obtain a lower bound of learning error which can be achieved without risk of over-fitting.

Functional Data Analysis: A statistical approach where multivariate data are treated as functions instead of discrete vectors.

Least Squares Support Vector Machine: A least squares modification of the Support Vector Machine which leads into solving a linear set of equations. Also bears close resemblance to Gaussian Processes.

Machine Learning: An area of Artificial Intelligence dealing with adaptive computational methods such as Artificial Neural Networks and Genetic Algorithms.

Over-Fitting: A common problem in Machine Learning where the training data can be explained well but the model is unable to generalize to new inputs. Over-fitting is related to the complexity of the model: any data set can be modelled perfectly with a model complex enough, but the risk of learning random features instead of meaningful causal features increases.

Support Vector Machine: A kernel based supervised learning method used for classification and regression. The data points are projected into a higher dimensional space where they are linearly separable. The projection is determined by the kernel function and a set of specifically selected support vectors. Training process involves solving a Quadratic Programming problem.

Variable Selection: Process where unrelated input variables are discarded from the data set. Variable selection is usually based on correlation or noise estimators of the input-output pairs and can lead into significant improvement in performance.

Functional Networks

Oscar Fontenla-Romero
University of A Coruña, Spain

Bertha Guijarro-Berdiñas
University of A Coruña, Spain

Beatriz Pérez-Sánchez
University of A Coruña, Spain

INTRODUCTION

Functional networks are a generalization of neural networks, which is achieved by using multiargument and learnable functions, i.e., in these networks the transfer functions associated with neurons are not fixed but learned from data. In addition, there is no need to include parameters to weigh links among neurons since their effect is subsumed by the neural functions. Another distinctive characteristic of these models is that the specification of the initial topology for a functional network could be based on the features of the problem we are facing. Therefore knowledge about the problem can guide the development of a network structure, although on the absence of this knowledge always a general model can be used.

In this article we present a review of the field of functional networks, which will be illustrated with practical examples.

BACKGROUND

Artificial Neural Networks (ANN) are a powerful tool to build systems able to learn and adapt to their environment, and they have been successfully applied in many fields. Their learning process consists of adjusting the values of their parameters, i.e., the weights connecting the network's neurons. This adaptation is carried out through a learning algorithm that tries to adjust some training data representing the problem to be learnt. This algorithm is guided by the minimization of some error function that measures how well the ANN is adjusting the training data (Bishop, 1995). This process is called parametric learning. One of the most popular neural network models are Multilayer Perceptrons (MLP) for which many learning algorithms can be used: from the brilliant backpropagation (Rumelhart, Hinton & Willian, 1986) to the more complex and efficient Scale Conjugate Gradient (Möller, 1993) or Levenberg-Marquardt algorithms (Hagan & Menhaj, 1994).

In addition, also the topology of the network (number of layers, neurons, connections, activation functions, etc.) has to be determined. This is called *structural learning* and it is carried out mostly by trial and error.

As a result, there are two main drawbacks in dealing with neural networks:

1. The resulting function lacks of the possibility of a physical or engineering interpretation. In this sense, Neural Networks act as black boxes.
2. There is no guarantee that the weights provided by the learning algorithm correspond to a global optimum of the error function, it can be a local one.

Models like Generalized Linear Networks (GLN) present an unique global optimum that can be obtained by solving a set of linear equations. However, its mapping function is limited as this model consists of a single layer of adaptive weights (w_j) to produce a linear combination of non linear functions (ϕ_j):

$$y(\mathbf{x}) = \sum_{j=0}^{M} w_j \phi_j(\mathbf{x}).$$

Some other popular models are Radial Basis Function Networks (RBF) whose hidden units use distances to a prototype vector (μ_j) followed by a transformation with a localized function like the Gaussian:

.

$$y(\mathbf{x}) = \sum_{j=0}^{M} w_j \, {}_j(\mathbf{x}) = \sum_{j=0}^{M} w_j \exp\left(-\frac{-\|\boldsymbol{\mu} - {}_j\|^2}{2\sigma_j^2} \right)$$

The resulting architecture is more simple than the one of the MLP, therefore reducing the complexity of structural learning and propitiating the possibility of physical interpretation. However, they present some other limitations like their inability to distinguish non significant input variables (Bishop, 1995), to learn some logic transformations (Moody & Darken, 1989) or the need of a large number of nodes even for a linear map if precision requirement is high (Youssef, 1993).

Due to these limitations, there have been appearing some models that extend the original ANN, such as, fuzzy neural networks (Gupta & Rao, 1994), growing neural networks, or probabilistic neural networks (Specht, 1990). Nowadays, the majority of these models still act as black boxes. Functional networks (Castillo, 1998, Castillo, Cobo, Gutiérrez, & Pruneda, 1998), a relatively new extension of neural networks, take into account the functional structure and properties of the process being modeled, that naturally determine the initial network's structure. Moreover, the estimation of the network's weights it is often based on an error function that can be minimized by solving a system of linear equations, therefore conducting faster to an unique and global solution.

DESCRIPTION OF FUNCTIONAL

NETWORKS

Functional networks (FN) are a generalization of neural networks, which is achieved by using multiargument and learnable functions (Castillo, 1998, Castillo, Cobo, Gutiérrez, & Pruneda, 1998), i.e., the shape of the functions associated with neurons are not fixed but learned from data. In this case, it is not necessary to include weights to ponder links among neurons since their effect is subsumed by the neural functions. Figure 1 shows an example of a general FN for $I=N_0$ explanatory variables.

Functional networks consist of the following elements:

a. Several layers of storing units (represented in Figure 1 by small filled circles). These units are used for the storage of both the input and the output of the network, or to storage intermediate information (see units $y_i^{(k)}$ in Figure 1).

b. One or more layers of functional units or neurons (represented by open circles with the name of each of the functional units inside). These neurons include a function that can be multivariate and that can have as many arguments as inputs. These arguments, and therefore the form of the neural functions, are learnt during training. By applying their functions, neurons evaluate a set of input

Figure 1. Generalized model for functional networks
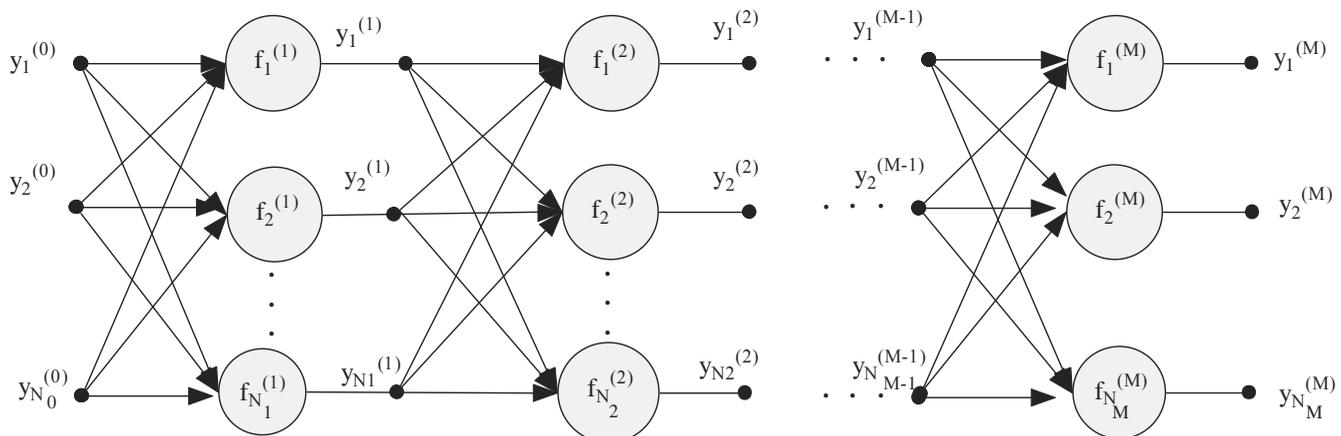
values in order to return a set of output values to the next layer of storing units. In this general model each neural function $f_i^{(m)}$ is defined as the following composition:

$$f_i^{(m)}\left(y_1^{(m-1)},...,y_{N_{m-1}}^{(m-1)}\right)=$$

$$g_i^{(m)}\left(h_{i1}^{(m-1)}\left(y_1^{(m-1)}\right),...,h_{iN_{m-1}}^{(m)}\left(y_{N_{m-1}}^{(m-1)}\right)\right)$$

where the superscript *(m)* is the number of layer. The functions $g_i^{(m)}$ are known and fixed before training, for example to be the sum or product. In contrast, functions $h_{ij}^{(m)}$ are lineal combinations of other known functions ϕ_{iq} (for example, polynomials, cosines, etc.), i.e. $h_{ij}^{(m)}\left(y_j^{(m-1)}\right)=\sum_{z=1}^{n_{ij}^{(m)}}a_{ijz}^{(m)}\phi_{ijz}^{(m)}\left(y_j^{(m-1)}\right)$ where the co-efficients $a_{ijz}^{(m)}$ implied in this linear combination are the model parameters to be learned. As can be observed, MLPs, GLNs and RBFs are particular cases of this generalized model.

c. A set of directed links that connect the functional units and the storing units. These connections indicate the direction of the flow of information. The general FN in Figure 1 does not have arrows that converge in the same storing unit, but if it did, this would indicate that the neurons from which they emanate must produce identical outputs. This is an important feature of FNs that is not available for neural networks. These converging arrows represent constraints which can arise from physical and/or theoretical characteristics of the problem under consideration.

Learning in Functional Networks

Functional networks combine knowledge about the problem to determine the network, and training data to estimate the unknown neural functions. Therefore, in contradistinction to neural networks, FNs include two types of learning:

1. *Structural learning*. The specification of the initial topology for a FN can be based on the features of the problem we are facing (Castillo, Cobo, Gutiérrez, & Pruneda, 1998). Usually knowledge about the problem can be used in order to develop a network structure. An important feature of FN is that they allow managing functional restric-

tions determined by some known properties of the model to be estimated. These restrictions can be representing by forcing the outputs of some neurons to coincide in a unique storage unit. Later on, the network can be translated into a system of functional equations that usually can be simplified in order to obtain a more simple but equivalent architecture. Finally, on the absence of knowledge about the problem always the general model, shown in Figure 1, can be used.

2. *Parametric learning*. This second stage refers to the estimation of the neuron's functions. Often these neural functions are considered to be lineal combinations of functional families, and therefore the parametric learning consists of estimating both the arguments of the neural functions and the parameters of the lineal combination using the available training data. It is important to remark that this type of learning generalizes the idea of estimating the weights of a neural network.

An Example Of A Functional Network

In this section the use of FNs is illustrated by means of an artificial simple example. Let's suppose a problem of engine diagnosis for which three continuous variables (x='vibrations', y='oil density', z='temperature') are being monitored. The problem is to estimate the probability P of a given diagnosis based on these variables, i.e., $P(x, y, z)$. Moreover, we know that the information provided by the monitored variables is accumulative. Therefore, it is possible, for example, to calculate first the probability $P_1(x, y)$ of a diagnosis based on only variables x and y, and later on when variable z is available combine the value provided by P_1 with the new information z to obtain $P(x, y, z)$. That is, there exist some functions such as:

$$P(x, y, z) \equiv F[P_1(x, y), z] = K[P_2(y, z), x] = L[P_3(x, z), y] \tag{1}$$

This situation suggests the structure of the FN shown in Figure 2a, where *I* is the identity function. The coincident connections in the store output unit, or equivalently eq. 1, establish strong restrictions about the functions P_1, P_2, P_3, *F*, *K*, *L*. The use of methods for functional equations allows to deal with eq. 1 in order to obtain the corresponding functional conditions from which it is possible to derive a new equation for

Figure 2. Functional network for the diagnosis example

a)   b)
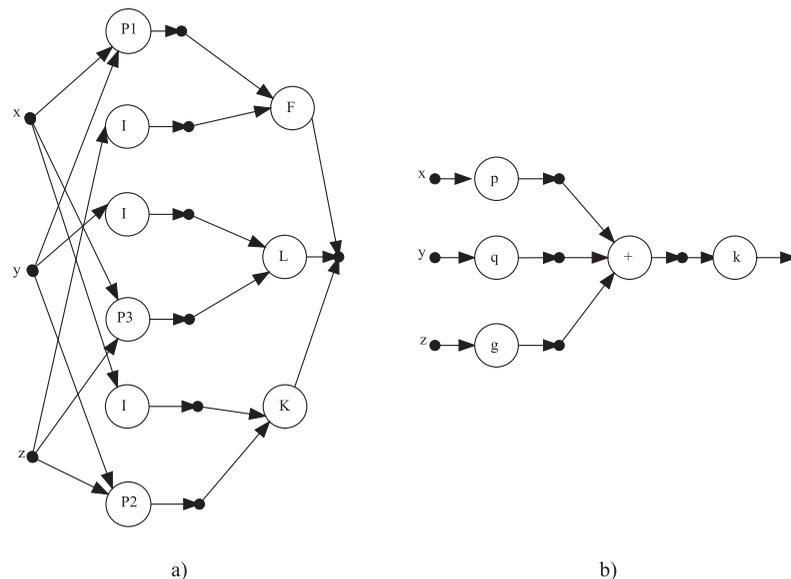
function *P*:

$$P(x, y, z) = k[p(x) + q(y) + g(z)].$$

This leads to the new more simple FN represented in Figure 2b which is equivalent to that of Figure 2a.

A Comparison Between Functional and Neural Networks

Although FNs are extensions of neural networks, there are some main features that distinguish both models:

1. Neural networks are derived only from data about the problem. However, FNs can also use knowledge about the problem to derive its topology, incorporating properties about the function to be modeled.
2. During learning in neural networks the shape of neural functions is fixed usually to be a sigmoid type function, and only the weights can be adapted. In FNs, neural functions are also learnt.
3. Neural functions that can be employed in neural networks are limited and belong to some known

family. Also, for each layer the same function is used for every neuron. In FNs any arbitrary function can be used for each neuron.
4. These functions can be multiargument and multivariate. In neural networks activation functions have only one argument (combination of several input data).
5. In FNs it is possible to force the output of some neurons to coincide by connecting them to the same storing unit. These connections are restrictions to the model that sometimes can be used to derive a more simple model.

Some Functional Network Models

In this section some typical FN models are presented, that let solving several real problems.

The Uniqueness Model

This is a simple but very powerful model for which the output *z* of the corresponding FN architecture can be written as a function of the inputs *x* and *y*,

$$z = F(x, y) = f_3^{-1}(f_1(x) + f_2(y)) \tag{2}$$

Uniqueness of Representation. For this model to have uniqueness of solution it is only required to fix the functions f_1, f_2, f_3 at a point (see explanation in Castillo, Cobo, Gutiérrez, & Pruneda, 1998).

Learning the model. Learning the function $F(x, y)$ in eq.2 is equivalent to learning the functions from a data set, $\{(x_i, y_i, z_i): j = 1,..., n\}$ where z is the desired output for the given inputs. To estimate f_1, f_2, f_3 we can employ the non-linear and linear methods:

1. The Non-Linear Method. We approximate each of the functions f_1, f_2, f_3^{-1} $z = F(x, y) = f_3^{-1}(f_1(x) + f_2(y))$ by considering them to be a linear combination of known functions from a given family (e. g., polynomial). Finally, the following sum of squared errors is minimized,

$$Q = \sum_{j=1}^{n} e_j^2 =$$

$$\sum_{j=1}^{n} \left(z_j - \sum_{k=1}^{m_3} a_{3k}\phi_{3k} \left(\sum_{i=1}^{m_1} a_{1i}\phi_{1i}(x_j) + \sum_{i=1}^{m_2} a_{2i}\phi_{2i}(y_j) \right) \right)^2$$

2. Linear Method. A simplification of the non-linear method can be done by considering the following equivalence:

$$z_j = f_3^{-1}\left(f_1(x_j) + f_2(y_j)\right) \Leftrightarrow$$
$$f_3(z_j) = f_1(x_j) + f_2(y_j); j = 1,...,n$$

Again the functions f_s can be approximated as a linear combination of known functions from a given family. Finally, the following sum of square errors,

$$Q = \sum_{j=1}^{n} e_j^2 =$$

$$\sum_{j=1}^{n} \left(\sum_{i=1}^{m_1} a_{1i}\phi_{1i}(x_j) + \sum_{i=1}^{m_2} a_{2i}\phi_{2i}(y_j) - \sum_{i=1}^{m_3} a_{3i}\phi_{3i}(z_j) \right)^2$$

can be minimized by solving a system of linear equations, where the unknowns are the coefficients a_{si} as it is demonstrated in (Castillo, Cobo, Gutiérrez, & Pruneda, 1998).

The Generalized Associativity Model

Figure 3a shows a generalized associativity FN of three inputs, where the nodes I represent the identity function. This model is based on the generalized associative property, that is, the output of this network can be obtained as a function of $G(x, y)$ and the input z, or

Figure 3. The generalized associativity functional network

a)   b)
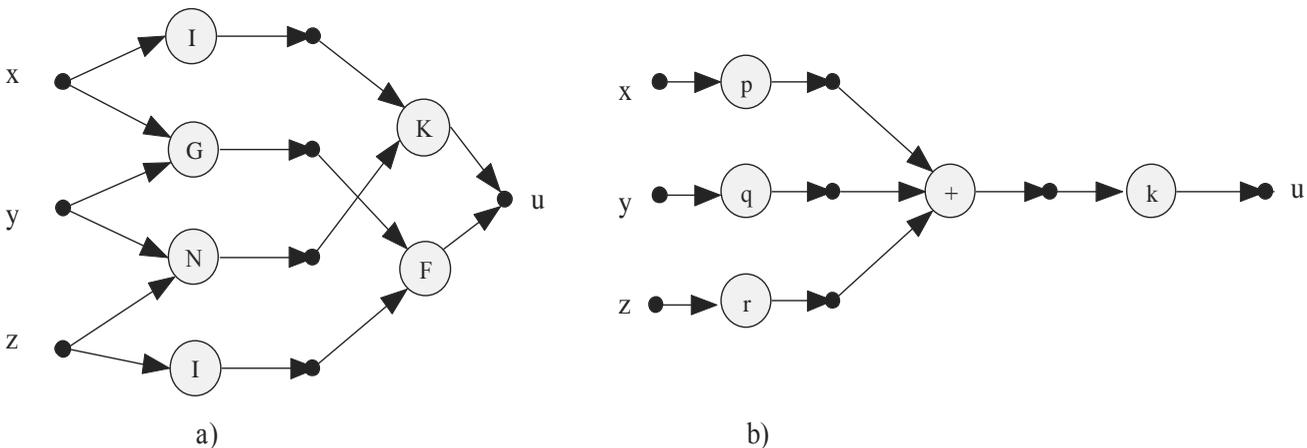
as a function of the input x and $N(y, z)$. This property is represented with the links convergent to the output node u, which leads to the functional equation

$$F[G(x, y), z] = K[x, N(y, z)] \qquad (3)$$

Simplification of the model. It can be shown that the general solution of eq. 3 is:

$$F(x, y) = k\big[f(x) + r(y)\big] G(x, y) = f^{-1}\big[p(x) + q(y)\big]$$
$$K(x, y) = k\big[p(x) + n(y)\big] N(x, y) = n^{-1}\big[q(x) + r(y)\big]$$
$$(4)$$

where f, r, k, n, p, q are arbitrary continuous and strictly monotonic functions. Substituting eq. 4 in eq. 3, the following result is obtained

$$F[G(x, y), z] = K[x, N(y, z)] = u = k[p(x) + q(y) + r(z)] \qquad (5)$$

Thus, the FN in Figure 3b is equivalent to the FN in Figure 3a.

Uniqueness of Representation. By employing functional equations for the generalized associativity model it can be demonstrated that uniqueness of solution requires fixing the functions k, p, q, r at a point (see Castillo, Cobo, Gutiérrez, & Pruneda, 1998).

Learning the model. The problem of learning the FN in Figure 3b involves estimating the functions k, p, q, r in eq. 5, that can be rewritten as:

$$k^{-1}(u) = p(x) + q(y) + r(z)$$

Being $\{(x_{1i}, x_{2i}, x_{3i}, x_{4i}) | i = 1,..,n\}$ with $(x_1, x_2, x_3, x_4 \equiv x, y, z, u)$ the observed training sample of size n we can define the error

$$e_i = \hat{p}(x_{1i}) + \hat{q}(x_{2i}) + \hat{r}(x_{3i}) - \hat{k}^{-1}(x_{4i}); i = 1,...,n$$

Figure 4. Separable functional network architecture

a)

b)
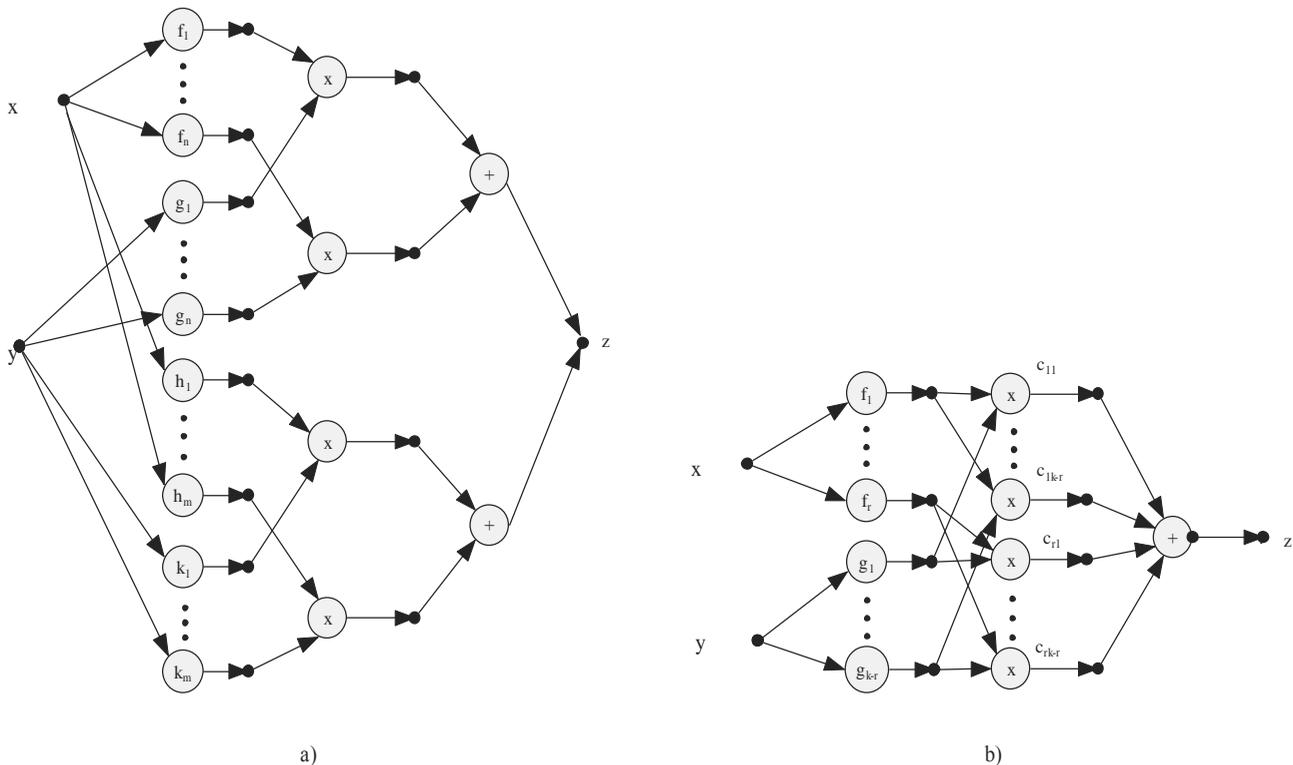
Suppose that each of the functions is a linear combination of known functions from given families (e.g. polynomial). Then, the sum of squared errors is defined as

$$Q = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(\sum_{k=1}^{4} \sum_{j=1}^{m_k} a_{kj} \phi_{kj}(x_{ki}) \right)^2 .$$

Employing the Lagrange multipliers technique, the minimum is obtained by solving the following system of linear equations:

$$\frac{\partial Q_\lambda}{\partial a_{kr}} = 2 \sum_{i=1}^{n} e_i \phi_{kr}(x_{ki}) + \lambda_k \phi_{kr}(\alpha_k) = 0; \forall k, r$$

$$\frac{\partial Q_\lambda}{\partial \lambda_k} = \sum_{j=1}^{m_s} a_{kj} \phi_{kj}(\alpha_k) - \beta_k = 0; \forall k,$$

where the unknowns are the multipliers $\lambda_1, ..., \lambda_4$ the coefficients in the set $\{a_{kj} \mid j = 1, ..., m_k; k = 1, 2, 3, 4\}$ which are the parameters of the FN.

The Separable Model

Consider the equation

$$z = F(x, y) = \sum_{i=1}^{n} f_i(x) r_i(y) = \sum_{j=1}^{m} h_j(x) k_j(y) ,$$

which can be written as

$$\sum_{i=1}^{k} f_i(x) g_i(y) = 0 \qquad (6)$$

where

$$f_i(x) = h_{i-n}(x)$$
$$g_i(y) = -k_{i-n}(y); i = n+1, ..., n+m$$

This suggests the FN in Figure 4a.

Simplification of the model. Assuming that $\{f_1(x), ... f_r(x)\}$, $\{g_{r+1}(x), ..., g_k(x)\}$ are two sets of linearly independent functions, the general solution of eq. 6

$$\sum_{i=1}^{k} f_i(x) g_i(y) = 0$$

is

$$f_j(x) = \sum_{j=1}^{r} a_{jk} f_k(x); j = r+1, ..., k,$$

$$g_s(y) = -\sum_{j=1}^{k-r} a_{js} g_{r+j}(y); s = 1, ..., r .$$

By replacing these terms in equation eq. 6 we obtain

$$z = F(x, y) = \sum_{i=1}^{r} \sum_{j=1}^{k-r} c_{ij} f_i(x) g_j(y) \qquad (7)$$

where c_{ij} are the parameters of the model, and which leads to the simplified FN in Figure 4b.

Uniqueness of Representation. In this case the uniqueness of representation is given without the need of fixing the implied functions at any point.

Learning the model. In this case a simple least squares method allows obtaining the optimal coefficients c_{ij} using the available data $\{(x_{0i}, x_{1i}, x_{2i}) | i = 1, ..., n\}$ with $(x_0, x_1, x_2 \equiv z, x, y)$. In this way, the error can be obtained as,

$$e_i = x_{0i} - \sum_{i=1}^{r} \sum_{j=1}^{s-r} c_{ij} f_i(x_{1i}) g_j(x_{2i}); i = 1, ..., n$$

Thus, to find the optimum coefficients we minimize the sum of squared errors

$$Q = \sum_{k=1}^{n} e_k^2 .$$

In this case, the parameters are not constrained by extra conditions, so the minimum can be obtained by solving the following system of linear equations, where the unknowns are the coefficients c_{ij}:

$$\frac{\partial Q}{\partial c_{pq}} = 2 \sum_{k=1}^{n} e_k f_p(x_{1k}) g_q(x_{2k}) = 0;$$

$$p = 1, ..., r; q = 1, ..., r - s$$

Examples of Applications

In this section, illustrative examples for two different models of FN are presented. These models were ap-

plied to a regression and a classification problem. In all cases, the functions of each layer were approximated by considering a linear combination of known functions from a polynomial family.

Classification Problem

The first example shows the performance of a FN solving a classification problem: the Wine data set. This database can be obtained from the UCI Machine Learning Repository[1]. The aim of this problem is to determine the origin of wines using a chemical analysis of 13 continuous attributes. The set contains 178 instances that must be classified in three different classes.

For this case, the Separable Model (Figure 4) with three output units was employed. Moreover, its performance is compared to other standard methods: a Multilayer Perceptron (MLP), a Radial Basis Function Network (RBF) and Support Vector Machines (SVM).

Figure 5 shows the comparative results. The first subfigure contains the mean accuracy obtained using a leaving-one-out cross-validation method. As can be observed, the FN obtains a very good performance for

Figure 5. Accuracy and training time obtained by different models for the wine data set
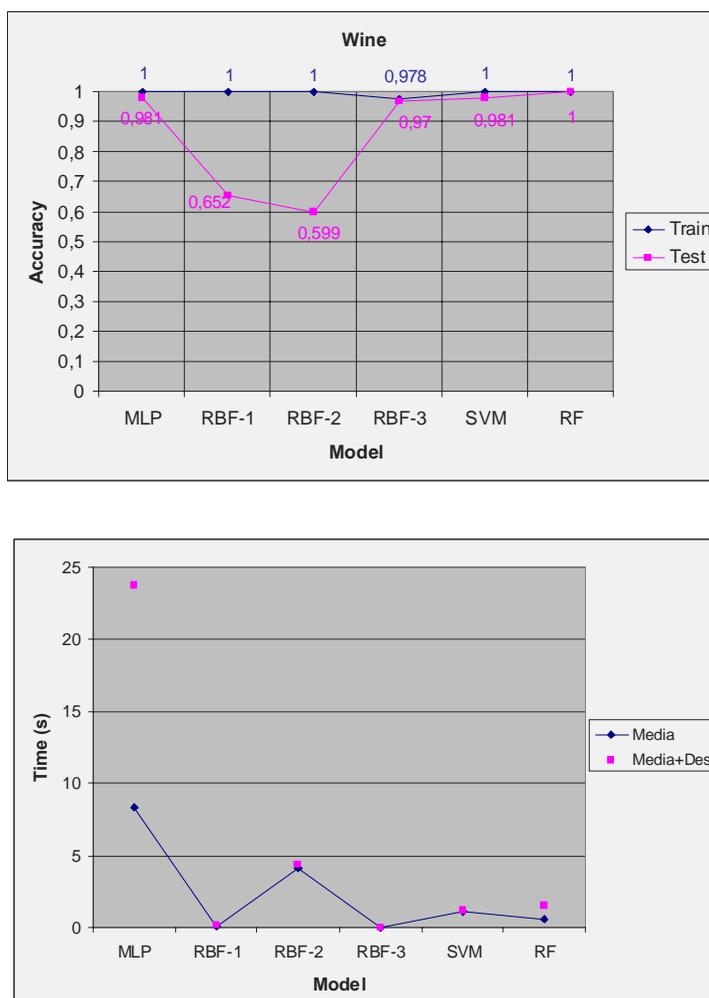
the test set. Regarding the time required for the learning process, the second subfigure shows that the FNs are comparable with the other methods.

Regression Problem

In this case the aim of the network is to predict the failure shear effort in concrete beams based on several geometrical, longitudinal and transversal parameters of the beam (Alonso-Betanzos, Castillo, Fontenla-Romero, & Sánchez-Maroño, 2004).

A FN, corresponding to the Associative Model (Figure 3), and also a MLP were trained employing a ten-fold cross-validation, running 30 simulations using different initial parameter values. A set with 12 samples was kept for further validation of the trained systems. The mean normalized Mean Squared Errors over 30 simulations obtained by the FN was 0.1789 and 0.8460 for test and validation, respectively, while the MLP obtained 0.1361 and 2.9265.

FUTURE TRENDS

Functional networks are being successfully employed in many different real applications. In engineering problems they have been applied, for instance, for surface reconstruction (Iglesias, Gálvez, & Echevarría, 2006). Other works have used these networks for recovering missing data (Castillo, Sánchez-Maroño, Alonso-Betanzos, & Castillo, 2003) and for general regression and classification problems (Lacruz, Pérez-Palomares & Pruneda, 2006) .

Another recent research line is related to the investigation of measures of fault tolerance (Fontenla-Romero, Castillo, Alonso-Betanzos, & Guijarro-Berdiñas, 2004), in order to develop new learning methods.

CONCLUSION

This article presents a review of functional networks. Functional networks are inspired by neural networks and functional equations. This model offers all the advantages of ANNs, such as noise tolerance and generalisation capacity, adding new advantages. One of them is the possibility to use knowledge about the problem to be modeled to derive the initial network topology, thus resulting on a model that can be physical

or engineering interpreted. Another main advantage is that the initially proposed model can be simplified, using functional equations, and learnt by solving a system of linear equations, which speeds the learning process and avoid it to be stuck in a local minimum. Finally, the shape of neural function does not have to be fixed, but they can be fitted from data during training, therefore widening the modeling ability of the network.

REFERENCES

Alonso-Betanzos, A., Castillo, E., Fontenla-Romero, O., & Sánchez-Maroño, N. (2004). Shear Strength Prediction using Dimensional Analysis and Functional Networks. *Proceedings of European Symposium on Artificial Neural Networks*, 251-256

Bishop, C.M. (1995). *Neural Networks for pattern recognition.* Oxford University Press.

Castillo, E. (1998). Functional networks. *Neural Processing Letters*, 7, 151-159.

Castillo, E., Cobo, A., Gutiérrez, J., & Pruneda R. (1998). Functional networks with applications. A neural-Based Paradigm. Kluwer Academic Publishers.

Castillo, E., & Gutiérrez, J.M. (1998). A comparison of functional networks and neural networks. *Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing,* 439-442

Castillo, E., Iglesias, A., & Ruiz-Cobo, R. (2004). Functional Equations in Applied Sciences. Elsevier.

Castillo, E., Sánchez-Maroño, N., Alonso-Betanzos, A., & Castillo, C. (2003). Recovering missing data with Functional and Bayesian Networks. *Lecture Notes in Computer Science*, 2687, part II, 489-496.

Fontenla-Romero, O., Castillo, E., Alonso-Betanzos, A., & Guijarro-Berdiñas, B. (2004). A measure of fault tolerance for functional networks. *Neurocomputing*, 62, 327-347.

Gupta, M., & Rao, D. (1994). On the principles of fuzzy neural networks. *Fuzzy Sets and Systems*, 61, 1-18.

Hagan, M.T. & Menhaj, M. (1994). Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), 989-993.

Fuzzy Approximation of DES State

Juan Carlos González-Castolo
CINVESTAV Unidad Guadalajara, Mexico

Ernesto López-Mellado
CINVESTAV Unidad Guadalajara, Mexico

INTRODUCTION

State estimation of dynamic systems is a resort often used when only a subset of the state variables can be directly measured; observers are the entities computing the system state from the knowledge of its internal structure and its (partially) measured behaviour. The problem of discrete event systems (DES) estimation has been addressed in (Ramirez, 2003) and (Giua 2003); in these works the marking of a Petri net (PN) model of a partially observed event driven system is computed from the evolution of its inputs and outputs.

The state of a system can be also inferred using the knowledge on the duration of activities. However this task becomes complex when, besides the absence of sensors, the durations of the operations are uncertain; in this situation the observer obtains and revise a belief that approximates the current system state. Consequently this approach is useful for non critical applications of state monitoring and feedback in which an approximate computation is allows.

The uncertainty of activities duration in DES can be handled using fuzzy PN (FPN) (Murata, 1996), (Cardoso, 1999), (Hennequin, 2001), (Pedrycz, 2003), (Ding, 2005); this PN extension has been applied to knowledge modelling (Chen, 1990), (Koriem, 2000), (Shen, 2003), planning (Cao, 1996), reasoning (Gao, 2003) and controller design (Andreu, 1997), (Leslaw, 2004).

In these works the proposed techniques include the computation of imprecise markings; however the class of models dealt does not include strongly connected PN for the modelling of cyclic behaviour. In this article we address the problem of state estimation of DES for calculating the fuzzy marking of a Fuzzy Timed Petri Net (FTPN); for this purpose a set of matrix expressions for the recursive computing the current fuzzy marking is developed. The article focuses on FTPN whose structure is a Marked Graph (called Fuzzy Timed Marked Graph -FTMG) because it allows showing intuitively the problems of the marking estimation in exhibiting cyclic behaviour.

BACKGROUND

Possibility Theory

In theory of possibility, a fuzzy set $\tilde{a}$ is used for delimiting ill-known values or for representing values characterized by symbolic expressions. The set is defined as $\tilde{a} = (a_1, a_2, a_3, a_4)$ such that $a_1, a_2, a_3, a_4 \in \mathbb{R}$, $a_1 \leq a_2$ and $a_3 \leq a_4$. The fuzzy set $\tilde{a}$ delimits the run time as follows:

- The values $\tilde{\tau}_b, \tilde{\tau}_a$ in the ranges (a_1, a_2), (a_3, a_4), respectively, indicate that the activity is possibly executed with $\alpha(\tau) \in (0,1)$. When $\tau \in \tilde{\tau}_b$ the function $\alpha(\tau)$ grows towards 1, which means that the possibility of stopping increases. When $\tau \in \tilde{\tau}_a$, the membership function $\alpha(\tau)$ decreases towards 0, representing that there is a reduction of the possibility of stopping.
- The values $(0, a_1]$ mean that the activity is running.
- The values $[a_4, +\infty)$ mean that the activity is stopped.
- The values $\tilde{\tau}_a \in [a_2, a_3] \mid a_2 \leq a_3$ represent full possibility that is $\alpha(\tau) = 1$, this represents that it is certain that the activity is stopped.
- The support of $\tilde{a}$ is the range $\tau \in [a_1, a_4]$ where $\alpha_{\tilde{a}}(\tau) > 0$.

A fuzzy set $\tilde{a}$ is referred indistinctly by the function $\alpha(\tau)$ or the characterization (a_1, a_2, a_3, a_4). For simplicity, in this work the fuzzy possibility distribution of the time is described with trapezoidal or triangular forms. For example, Fig.1 shows the fuzzy set that

Figure 1. Fuzzy set
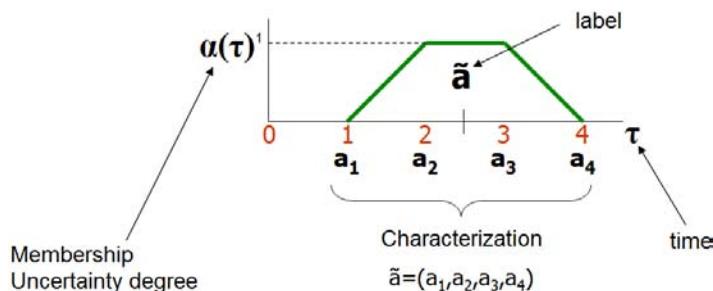
it is represents in natural language: "the activity will stop about 2.5".

Fuzzy extension principle. The fuzzy extension principle plays a fundamental role because we can extend functions defined on crisp sets to functions on fuzzy sets. An important application of this principle is a mechanism to operate arithmetically with fuzzy numbers.

Definition. *Let $X_1,\ldots,X_n$ be crisp sets and let f a function such $f : X_1 \times \ldots \times X_n \to Y$. If $\tilde{a}_1,\ldots,\tilde{a}_n$ are fuzzy sets on $X_1,\ldots,X_n$, respectively, then $f(\tilde{a}_1,\ldots,\tilde{a}_n)$ is the fuzzy set on Y such that:*

$$f(\tilde{a}_1,\ldots,\tilde{a}_n) = \cup_{(x_1,\ldots,x_n)\in(X_1\times\ldots\times X_n)}\left\{\alpha_{\tilde{a}_1}(x_1)\wedge\ldots\wedge\alpha_{\tilde{a}_n}(x_n)/f(x_1,\ldots,x_n)\right]\right\}$$

If $\tilde{b} = f(\tilde{a}_1,\ldots,\tilde{a}_n)$ then $\tilde{b}$ is the fuzzy set on Y such that:

$$\tilde{b}(y) = \vee_{(x_1,\ldots,x_n)\in(X_1\times\ldots\times X_n),f(x_1,\ldots,x_n)=y}\left[\alpha_{\tilde{a}_1}(x_1)\wedge\ldots\wedge\alpha_{\tilde{a}_n}(x_n)\right]$$

The fuzzy set was characterized as:

$$\tilde{a} = \left\{\alpha_{\tilde{a}}(x_1)/x_1,\ldots,\alpha_{\tilde{a}}(x_n)/x_n\right\}.$$

With the extension principle we can define a simplified fuzzy sets addition operation.

Definition. *Let $\tilde{a} = (a_1,a_2,a_3,a_4)$ and $\tilde{b} = (b_1,b_2,b_3,b_4)$ be two trapezoidal fuzzy sets. The fuzzy sets addition operation is: $\tilde{a}\oplus\tilde{b} = (a_1+b_1, a_2+b_2, a_3+b_3, a_4+b_4)$ (Klir, 1995).*

Definition *The intersection and union of fuzzy sets are defined in terms of min and max operators.*

$$(\tilde{a}\cap\tilde{b}) = \min(\tilde{a},\tilde{b}) = \min\left(\alpha_{\tilde{a}}(\tau),\alpha_{\tilde{b}}(\tau)\right)|\ \tau\in support\_of\_\tilde{a}\wedge\tilde{b}$$

and

$$(\tilde{a}\cup\tilde{b}) = \max(\tilde{a},\tilde{b}) = \max\left(\alpha_{\tilde{a}}(\tau),\alpha_{\tilde{b}}(\tau)\right)|\ \tau\in support\_of\_\tilde{a}\vee\tilde{b}$$

We used these operators, intersection and union, as a t-norm and a s-norm, respectively.

Definition *The distribution of possibility before and after $\tilde{a}$ are the fuzzy sets $\tilde{a}^b = (-\infty, a_2, a_3, a_4)$ and $\tilde{a}^a = (a_1, a_2, a_3, +\infty)$ respectively; they are defined in (Andreu, 1997) as a function $\alpha_{(-\infty,\tilde{a}]}(\tau) = \sup_{\tau'\geq\tau}\alpha(\tau')$ and $\alpha_{(\tilde{a},+\infty]}(\tau) = \sup_{\tau'\leq\tau}\alpha(\tau')$, respectively.*

Petri Nets Theory

Definition. *An ordinary PN structure G is a bipartite digraph represented by the 4-tuple $G = (P,T,I,O)$ where $P = \{p_1, p_2,\ldots, p_n\}$ and $T = \{t_1, t_2,\ldots, t_m\}$ are finite sets of vertices called respectively places and transitions, $I(O): P\times T \to \{0,1\}$ is a function representing the arcs going from places to transitions (transitions to places).*

Pictorially, places are represented by circles, transitions are represented by rectangles, and arcs are depicted as arrows. The symbol $\bullet t_j (t_j \bullet)$ denotes the set

of all places p_i such that $I(p_i, t_j) \neq 0$ $(O(p_i, t_j) \neq 0)$. Analogously, $\bullet p_i$ $(p_i \bullet)$ denotes the set of all transitions t_j such that $O(p_i, t_j) \neq 0$ $(I(p_i, t_j) \neq 0)$.

The pre-incidence matrix of G is $C^- = \left[c_{ij}^- \right]$ where $c_{ij}^- = I(p_i, t_j)$; the post-incidence matrix of G is $C^+ = \left[c_{ij}^+ \right]$ where $c_{ij}^+ = O(p_i, t_j)$; the incidence matrix of G is $C = C^+ - C^-$.

A marking function $M : P \to \mathbb{Z}^+$ represents the number of tokens or marks (depicted as dots) residing inside each place. The marking of a *PN* is usually expressed as an n-entry vector.

Definition. *A Petri Net system or Petri Net (PN) is the pair N = (G, M_0), where G is a PN structure and M_0 is an initial token distribution.*

In a *PN* system, a transition t_j is enabled at the marking M_k if $\forall p_i \in P, M_k(p_i) \geq I'(p_i, t_j)$; an enabled transition t_j can be fired reaching a new marking M_{k+1} which can be computed using the *PN* state equation:

$$M_{k+1} = M_k + C^+ v_k - C^- v_k \qquad (1)$$

where $v_k(i) = 0, i \neq j, v_k(j) = 1$.

The reachability set of a *PN* is the set of all possible reachable marking from M_0 firing only enabled transitions; this set is denoted by $R(G, M_0)$.

A structural conflict is a *PN* sub-structure in which two or more transitions share one or more input places; such transitions are simultaneously enabled and the firing of one of them may disable the others, Fig.3(b).

Definition. *A transition $t_k \in T$ is live, for a marking M_0, if $\forall M_k \in R(G, M_0)$, $\exists M_n \in R(G, M_0)$ such that t_k is enabled*

$$\left(M_n \xrightarrow{t_k} \right).$$

A PN is live if all its transitions are live.

Definition. *A PN is said 1-bounded, or safe, for a marking M_0, if $\forall p_i \in P$ and $\forall M_j \in R(G, M_0)$, it holds that $M_j(p_i) \leq 1$.*

In this work we deal with live and safe *PN*.

Definition. *A p-invariant Y_i (t-invariant X_i) of a PN is a positive integer solution of the equation $Y_i^T C = 0$ (CX$_i$ = 0). The support of the p-invariant Y_i (t-invariant X_i) is the set $\|Y_i\| = \{ p_j \mid Y_i(p_j) \neq 0 \}$ $\left(\|X_i\| = \{ t_j \mid X_i(t_j) \neq 0 \} \right)$*

Definition. *Let Y_i a p-invariant of a Petri net (G, M_0), $\|Y_i\|$ the support of Y_i, then the induced subnet by Y_i is*

$$PC_i = (P_i = \|Y_i\|, T_i = \cup t_k \in \bullet p_j, t_l \in p_j \bullet$$
$$\mid p_j \in \|Y_i\|, I_i, O_i)$$

named p-component, where

$$I_i = P_i \times T_i \cap I \ , \ O_i = P_i \times T_i \cap O.$$

Definition. *Let X_i be a t-invariant of a PN, and $\|X_i\|$ be the support of X_i, then the induced subnet by X_i is*

$$TC_i = \left(P_i = \{ \cup p_k \in \bullet t_j, p_l \in t_j \bullet \right.$$
$$\left. \mid t_j \in \|X_i\| \}, T_i = \|X_i\|, I_i, O_i \right).$$

named t-component.

$$I_i = P_i \times T_i \cap I \quad and \quad O_i = P_i \times T_i \cap O.$$

Definition. *A invariant Z_i is minimal if no invariant Z_j satisfies $\|Z_j\| \subset \|Z_i\|$, where Z_i, Z_j are p-invariants or t-invariants and $\forall z \in Z_i : z \geq 0$.*

Definition. *Let $Z = \{ Z_1, \ldots, Z_q \}$ be the set of minimal invariants (Silva, 1982) of a PN, then Z is called the invariants base. The cardinality of Z is represented as $|Z|$.*

FUZZY TIMED PETRI NETS

Basic Operators

We introduce first some useful operators.

Definition. *In order to get the fuzzy set between $\tilde{f}$ and $\tilde{g}$, the lmax function is defined as:*

$$lmax\left(\tilde{f},\tilde{g}\right)= min\left(\tilde{f}^{a},\tilde{g}^{b}\right) \tag{2}$$

Definition. *The latest (earliest) operation selects the latest (earliest) fuzzy set among n fuzzy sets; they are calculated as follows:*

$$latest\left(\tilde{f}_{1},\ldots,\tilde{f}_{n}\right)=$$
$$min\left(max\left(\tilde{f}_{1}^{b},\ldots,\tilde{f}_{n}^{b}\right),min\left(\tilde{f}_{1}^{a},\ldots,\tilde{f}_{n}^{a}\right)\right) \tag{3}$$

$$earliest\left(\tilde{f}_{1},\ldots,\tilde{f}_{n}\right)=$$
$$min\left(min\left(\tilde{f}_{1}^{b},\ldots,\tilde{f}_{n}^{b}\right),max\left(\tilde{f}_{1}^{a},\ldots,\tilde{f}_{n}^{a}\right)\right) \tag{4}$$

Definition. *The fuzzy_conjugation-operator is defined as* $arg1 \overset{oper}{\bullet} arg2$, *where arg1, arg2 are arguments that can be matrices of fuzzy sets;* $\cdot$ *is the fuzzy and operation and oper is any operation referred as, +, -, latest, min, etc. For some row i = 1,...m and some column j = 1,... n) the products* $and\left(\tilde{f}_{ik},\tilde{g}_{kj}\right)| k=1,\ldots,r$ *are computed as* $oper\left(and\left(\tilde{f}_{ik},\tilde{g}_{kj}\right)\right)$. *For example:*

$$\begin{bmatrix} \tilde{f}_{11} & \cdots & \tilde{f}_{1r} \\ \vdots & \ddots & \\ \tilde{f}_{m1} & & \tilde{f}_{mr} \end{bmatrix} \overset{+}{\bullet} \begin{bmatrix} \tilde{g}_{11} & \cdots & \tilde{g}_{1n} \\ \vdots & \ddots & \\ \tilde{g}_{r1} & & \tilde{g}_{rn} \end{bmatrix} =$$

$$\begin{bmatrix} \sum_{k=1}^{r}\tilde{f}_{1k}\tilde{g}_{k1} & \cdots & \sum_{k=1}^{r}\tilde{f}_{1k}\tilde{g}_{kn} \\ \vdots & \ddots & \\ \sum_{k=1}^{r}\tilde{f}_{mk}\tilde{g}_{k1} & & \sum_{k=1}^{r}\tilde{f}_{mk}\tilde{g}_{kn} \end{bmatrix}$$

Formalism Description of the FTPN

Definition. *A fuzzy timed Petri net structure is a 3-tuple* $FTPN = \left(N,\Gamma,\xi\right)$; *where N = (G, M_0) is a PN,* $\Gamma = \{\tilde{a}_1,\tilde{a}_2,\ldots,\tilde{a}_n\}$ *is a collection of fuzzy sets,* $\xi : P \to \Gamma$ *is a function that associates a fuzzy set* $\tilde{a}_i \in \Gamma$ *to each place* $p_i \in P$.

• Fuzzy timing of places

The fuzzy set $\tilde{a} = \left(a_1,a_2,a_3,a_4\right)$ Fig.2(b) represents the *static* possibility distribution $\alpha\left(\tau_a\right)\in\left[0,1\right]$ of the instant at which a token leaves a place $p \in P$, starting from the instant when p is marked. This set does not change during the FTPN execution.

• Fuzzy timing of tokens

The fuzzy set $\tilde{b} = \left(b_1,b_2,b_3,b_4\right)$ Fig.2(c) represents the *dynamic* possibility distribution $\beta\left(\tau_b\right)\in\left[0,1\right]$ associated to a token residing within a $p \in P$; it also represents the instant τ_b at which such a token leaves the place, starting from the instant when p is marked. $\tilde{b}$ is computed from $\tilde{a}$ every time the place is marked
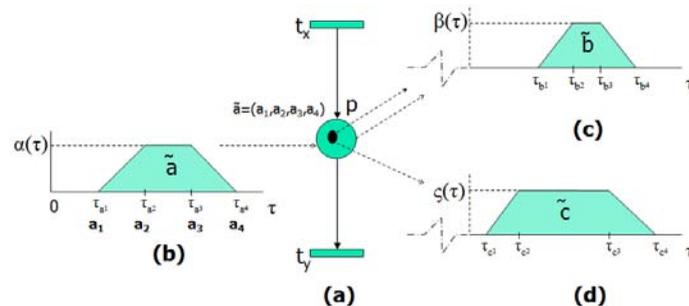
Figure 2. (a) Fuzzy timed Petri net. (b) The fuzzy set associated to places. (c) Fuzzy set to place or mark associated. (d) Fuzzy timestamp

F

during the marking evolution of the FTPN. A token begins to be available for enabling output transitions at $\beta\left(b_1\right)$. Thus $\tilde{b}^a=\left(b_1,b_2,b_3,+\infty\right)$ represents the possibility distribution of available tokens. The fuzzy set $\tilde{c}=\left(c_1,c_2,c_3,c_4\right)$, known as *fuzzy timestamp*, Fig.2(d) is a dynamic possibility distribution $\varsigma\left(\tau_c\right)\in\left[0,1\right]$ that represents the duration of a token within a place $p\in P$.

Enabling and Firing of Transitions

- Fuzzy enabling date

The fuzzy enabling date $e_{t_k}\left(\tau\right)$ of the transition t_k at the instant τ is a possibility distribution of the latest leaving instant among the leaving instants $\tilde{b}_{p_i}$ of all tokens within the $p_i\in\bullet t_k$, Fig.3(a).

$$e_{t_k}\left(\tau\right)=latest\left(\tilde{b}_{p_i}\right)\forall p_i\in\bullet t_k \tag{5}$$

The *latest* operation obtains the latest date in which the input places p_i to t_k have a token.

- Fuzzy firing date

The firing transition date $o_{t_k}\left(\tau\right)$ of a transition t_k is determined with respect to the set of transitions $\{t_j\}$ simultaneously enabled, Fig.3(b). This date, expressed as a possibility distribution, is computed as follows:

$$o_{t_k}\left(\tau\right)=min\left(e_{t_k}\left(\tau\right),earliest\left(e_{t_j}\left(\tau\right)\right)\forall t_k\in p_n\bullet;p_n\in\bullet t_j\right) \tag{6}$$

The *earliest* operation obtains the earliest date in which the transitions in a structural conflict are enabled.

- Fuzzy timestamp

For a given place p_s, the possibility distribution $\tilde{b}_{p_s}$ may be computed from a_{p_s} and the firing dates $o_{t_j}\left(\tau\right)$ of a $t_j\in\bullet p_s$ using the following expression:

$$\tilde{b}_{p_s}=lmax\left(o_{t_j}\left(\tau\right)\right)\oplus\tilde{a}_{p_s}\,\forall t_j\in\bullet p_s \tag{7}$$

The token do not disappear of $\bullet t$ and appear in $t\bullet$ instantaneously. The fuzzy timestamp $\tilde{c}_{p_s}$ is the time elapse possibility that a token is in a place $p_s\in P$. The possibility distribution $\tilde{c}_{p_s}$ is computed from the occurrence dates of both $\bullet p_s$ and $p_s\bullet$, see Fig.3(c).

$$c_{p_s}=lmax\left(earliest\left(o_{t_i}\left(\tau\right)\right),latest\left(o_{t_j}\left(\tau\right)\right)\right)\forall t_i\in\bullet p_s,t_j\in p_s\bullet \tag{8}$$

Actually, $\tilde{c}_{p_s}$ represent the fuzzy marking at the instant τ.

Matrix Formulation

Now, we reformulate the expressions (5),(6),(7) and (8) allowing a more general an compact representation.

$$\tilde{B}=\left(C^+\overset{lmax}{\bullet}\tilde{O}\right)\oplus\tilde{A} \tag{9}$$

$$\tilde{E}=\left[C^-\right]^T\overset{latest}{\bullet}\tilde{B} \tag{10}$$

$$\tilde{O}=\left[C^-\right]^T\overset{min}{\bullet}\left(C^-\overset{earliest}{\bullet}\tilde{E}\right) \tag{11}$$

Figure 3. (a) Conjunction transition. (b) Structural conflict. (c) Attribution and selection place
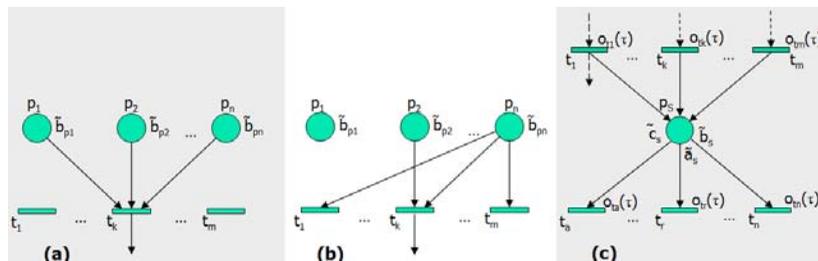
$$\tilde{C} = lmax\left(C^+ \overset{earliest}{\bullet} \tilde{O}, C^- \overset{latest}{\bullet} \tilde{O} \right) \tag{12}$$

where $\tilde{B}, \tilde{E}, \tilde{O}$ and $\tilde{C}$ denote vectors composed by $\tilde{b}_{p_s}, \tilde{e}_{t_k}, \tilde{o}_{t_k}, \tilde{c}_{p_s}$, respectively.

Modeling Example

Now we will illustrate the previous matrix formulation though a simple example.

Example

Consider the system shown in Fig.4(a); it consist of two cars, car1 and car2, which move along independent and dependent ways executing the set of activities Op={Right_car1, Right_car2, Charge_car1, Charge_car2, Left_car1,2 Discharge_car1,2}. The operation of the system is automated following the sequence described in the FTPN of Fig.4(b) in which the activities are associated to places. The ending time possibility $\tilde{a}_{p_i}$ for every activity is given in the model. We are considering that there are not sensors detecting the tokens in the system, thus the behavior is then analyzed through the estimated state.

a. Initial conditions: Initially, $M_0 = \{p_1\}$, therefore, the enabling date $e_{t_1}(\tau)$ of transitions t_1 is immediate, i.e., (0,0,0,0). Since $|\bullet t_1| = 1$, then $o_{t_1}(\tau) = e_{t_1}(\tau)$.
b. Matrix equations: For the obtained the fuzzy sets we solve (9)-(12) as follows:

$$\tilde{B} = \begin{bmatrix} \tilde{b}_{p_1} \\ \tilde{b}_{p_2} \\ \tilde{b}_{p_3} \\ \tilde{b}_{p_4} \\ \tilde{b}_{p_5} \\ \tilde{b}_{p_6} \end{bmatrix} = \begin{bmatrix} o_{t_5} \oplus \tilde{a}_{p_1} \\ o_{t_1} \oplus \tilde{a}_{p_2} \\ o_{t_1} \oplus \tilde{a}_{p_3} \\ o_{t_2} \oplus \tilde{a}_{p_4} \\ o_{t_3} \oplus \tilde{a}_{p_5} \\ o_{t_4} \oplus \tilde{a}_{p_6} \end{bmatrix} \tag{13}$$

$$\tilde{E} = \begin{bmatrix} e_{t_1} \\ e_{t_2} \\ e_{t_3} \\ e_{t_4} \\ e_{t_5} \end{bmatrix} = \begin{bmatrix} \tilde{b}_{p_1} \\ \tilde{b}_{p_2} \\ \tilde{b}_{p_3} \\ latest\left(\tilde{b}_{p_4}, \tilde{b}_{p_5}\right) \\ \tilde{b}_{p_6} \end{bmatrix} \tag{14}$$

$$\tilde{O} = \begin{bmatrix} o_{t_1} \\ o_{t_2} \\ o_{t_3} \\ o_{t_4} \\ o_{t_5} \end{bmatrix} = \begin{bmatrix} e_{t_1} \\ e_{t_2} \\ e_{t_3} \\ e_{t_4} \\ e_{t_5} \end{bmatrix} \tag{15}$$

Figure 4. (a) Two cars system. (b) Fuzzy timed Petri net model
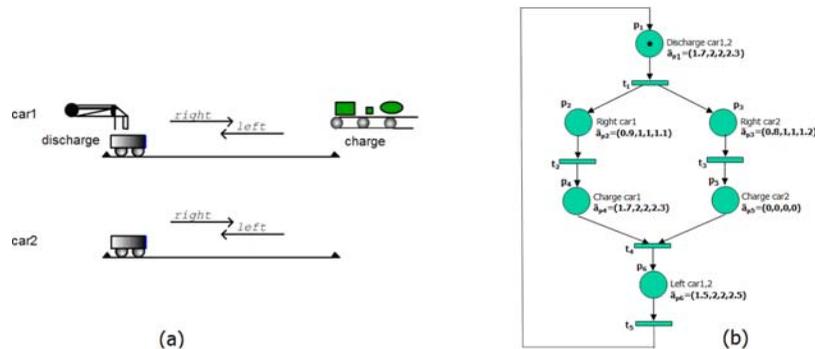
$$\tilde{C} = \begin{bmatrix} \tilde{c}_{p_1} \\ \tilde{c}_{p_2} \\ \tilde{c}_{p_3} \\ \tilde{c}_{p_4} \\ \tilde{c}_{p_5} \\ \tilde{c}_{p_6} \end{bmatrix} = \begin{bmatrix} lmax\left(o_{t_5}, o_{t_1}\right) \\ lmax\left(o_{t_1}, o_{t_2}\right) \\ lmax\left(o_{t_1}, o_{t_3}\right) \\ lmax\left(o_{t_2}, o_{t_4}\right) \\ lmax\left(o_{t_3}, o_{t_4}\right) \\ lmax\left(o_{t_4}, o_{t_5}\right) \end{bmatrix} \tag{16}$$

c. Firing t_1: When t_1 is fired, the token is removed from p_1; p_2 and p_3 get one token each one.

$$\tilde{B} = \begin{bmatrix} 0 & (0.9,1,1,1.1) & (0.8,1,1,1.2) & 0 & 0 & 0 \end{bmatrix}^T$$

The possibility sets $\tilde{b}_{p_2}$, $\tilde{b}_{p_3}$ coincide with $\tilde{a}_{p_2}$ and $\tilde{a}_{p_3}$, respectively.

d. Firing t_2: The fuzzy enabling time and the fuzzy occurrence time are computed by (14) and (15), respectively.

$$\tilde{O}, \tilde{E} = \begin{bmatrix} 0 & (0.9,1,1,1.1) & 0 & 0 & 0 \end{bmatrix}^T$$

The set $\tilde{c}_{p_2}$ is the possibility distribution of the time at which p_2 is marked. So, we computed (16).

$$\tilde{C} = \begin{bmatrix} 0 & (0,0,1,1.1) & 0 & 0 & 0 \end{bmatrix}^T$$

The set $\tilde{b}_{p_4}$ is the possibility distribution of the instant at which place p4 losses the token and it can be calculated by (13).

$$\tilde{B} = \begin{bmatrix} 0 & 0 & 0 & (2.6,3,3,3.4) & 0 & 0 \end{bmatrix}^T$$

e. Firing t_3: Again, using (14), (15) and (16) we obtain:

$$\tilde{O}, \tilde{E} = \begin{bmatrix} 0 & 0 & (0.8,1,1,1.2) & 0 & 0 \end{bmatrix}^T$$

$$\tilde{C} = \begin{bmatrix} 0 & 0 & (0,0,1,1.2) & 0 & 0 & 0 \end{bmatrix}^T$$

$$\tilde{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & (2.6,3,3,3.4) & 0 \end{bmatrix}^T$$

Figure 5(a) present the marking evolution of one cycle and some more steps. $\tilde{C}$ is represented by the dashed line and $\tilde{B}$ is represented by the shadowed area. Notice that $\tilde{O}$ coincide sometimes with $\tilde{B}$.

FUZZY STATE EQUATION

We analyzed equation (1) in order to obtain the fuzzy marking equation. $C^+ v_k$ provides information about the places that get tokens. Also, we must consider that in FTPN the transition firing possibility evolves continuously. The variation of $O\left(\tilde{\tau}_b\right)$ during $\tau \in \tilde{\tau}_b$ modifies the possibility of tokens residing in the output places of the firing transitions; thus the corresponding term to v_k in FTPN is rather a variation denoted by $\Delta_{\tilde{O}\left(\tilde{\tau}_b\right)}$; thus the marking variation is $C^+ \Delta_{\tilde{O}\left(\tilde{\tau}_b\right)}$. By a similar reasoning on the term $C^- v_k$ corresponds to $C^- \Delta_{\tilde{O}\left(\tilde{\tau}_a\right)}$ in FTPN. The operation $C^+ \Delta_{\tilde{O}\left(\tilde{\tau}_b\right)} - C^- \Delta_{\tilde{O}\left(\tilde{\tau}_a\right)}$ represents the possible marking change. Considering the marking after a time elapse $\Delta\tau$ we obtain:

$$M\left(\tau\right) = M\left(\tau - \Delta\tau\right) + C^+ \Delta_{\tilde{O}\left(\tilde{\tau}_b\right)} - C^- \Delta_{\tilde{O}\left(\tilde{\tau}_a\right)} \tag{17}$$

Here

$$\Delta_{\tilde{O}\left(\tilde{\tau}_b\right)} = \tilde{O}\left(\tau\right) - \tilde{O}\left(\tau - \Delta\tau\right) | \left(\tau - \Delta\tau, \tau\right) \in \tilde{\tau}_b$$

and

$$\Delta_{\tilde{O}\left(\tilde{\tau}_a\right)} = \tilde{O}\left(\tau - \Delta\tau\right) - \tilde{O}\left(\tau\right) | \left(\tau - \Delta\tau, \tau\right) \in \tilde{\tau}_a.$$

The marking possibility obtained in (17) can be greater than 1; then since FTPN are safe, we use the min function to obtain $M\left(\tau\right) \leq 1$.

The new marking is denoted by $\hat{M}\left(\tau\right)$, i.e.,

$$\hat{M}\left(\tau\right) = \min\left(M\left(\tau - \Delta\tau\right) + C^+ \Delta_{\tilde{O}\left(\tilde{\tau}_b\right)} - C^- \Delta_{\tilde{O}\left(\tilde{\tau}_a\right)}, \vec{1}\right) \tag{18}$$

where $\vec{1}$ is a n-entry vector containing 1 in each entry.

Initially $M\left(0\right) = M_0$. If $\tau \neq 0$ then (18) is solved in three steps:

- $\breve{M}\left(\tau\right) = C^+ \Delta_{\tilde{O}\left(\tilde{\tau}_b\right)} - C^- \Delta_{\tilde{O}\left(\tilde{\tau}_a\right)}$

- $M\left(\tau\right) = M\left(\tau - \Delta\tau\right) + \breve{M}\left(\tau\right)$

- $\hat{M}(\tau) = \min\left(M(\tau), \vec{1}\right)$

Remark.
If $\Delta_{\tilde{O}(\tilde{\tau}_b)}, \Delta_{\tilde{O}(\tilde{\tau}_a)} \in \{0,1\}$ the behaviour is that of an ordinary timed Petri net.

Example.
For the system shown in Fig.4, we obtained the marking in some instants. The initial marking is $M(0) = [1\ 0\ 0\ 0\ 0\ 0]$. The transition t_1 is firing at $\tau = 0^+$, therefore $M(0^+) = [0\ 1\ 1\ 0\ 0\ 0]$. For $\tau \in \left(0^+, 0.8\right)$ the marking does not change. For $\tau = 1$ we obtain:

$$\breve{M}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$M(1) = M(0.8) + \breve{M}(1) = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}^T$$

$$\hat{M}(1) = \min\left(M(1), \vec{1}\right) = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}^T$$

The marking evolution at some relevant instants is shown below:

t	0	0.8	1	2	2.8	3	3.2	4	5
$\hat{M}_{p_1}(t)$	1	0	0	0	0	0	0	0	1
$\hat{M}_{p_2}(t)$	0	1	1	0	0	0	0	0	0
$\hat{M}_{p_3}(t)$	0	1	1	0	0	0	0	0	0
$\hat{M}_{p_4}(t)$	0	0	1	1	1	1	0.5	0	0
$\hat{M}_{p_5}(t)$	0	0	1	1	1	1	0.5	0	0
$\hat{M}_{p_6}(t)$	0	0	0	0	0	0.66	1	1	1

Notice that during $\tau \in (0,5)$, $\hat{M}(\tau)$ coincides with the fuzzy timestamp; it is shown in Fig.5(a).

STATE APPROXIMATION OF THE FTPN

Marking Estimation

Definition. *The marking estimation Ξ in the instant τ is described by the function $\psi_{Y_i}(\tau) \in [0,1]$ which recognize the possible marked place $p_u \in \|Y_i\| \mid i \in \{1,...,|Y|\}$, among other possible places $p_v \in \|Y_i\| \mid v \neq u$. The function $\psi_{Y_i}(\tau)$ evaluates the minimal difference that exist*

among the bigger possibility $\hat{M}_u(\tau)$ that the token is in a place u and the possibility $\hat{M}_v(\tau)$ that the token is in any other place. The function $\psi_{Y_i}(\tau)$ is then calculated.

$$\psi_{Y_i}(\tau) = \min\left(\left|\hat{M}_{p_u}(\tau) - \hat{M}_{p_v}(\tau)\right|\right) \qquad (19)$$

such that $\forall \{p_u, p_v\} \in \|Y_i\|; v \neq u; Y_i \in Y$

Example.
The FTPN in Fig.4 has two p-invariants with supports $Y_1 = \|p_1, p_2, p_4, p_5\|$ and $Y_2 = \|p_1, p_3, p_5\|$. Figure 6 shows the fuzzy sets $\tilde{C}$ obtained from evolution of the marking in the p-component induced by Y_1, Fig.5(b).

Definition. *The state estimation S, at the instant τ is described by the function $s(\tau) \in [0,1]$, which determines the possible state of the system among other possible states; it is calculated by:*

$$s(\tau) = \min\left(\psi_{Y_i}(\tau)\right) \mid i = 1,...,|Y|; Y_i \in Y \qquad (20)$$

Discrete State From the FTPN

In order to obtain a possible discrete marking $\bar{M}(\tau)$ of the FTPN it is necessary to perform a "defuzzyfication" of $M(\tau)$. This can be accomplished taking into account the possible discrete marking $\bar{M}_i(\tau)$ of every p-component induced by Y_i. Before describing the procedure to obtain $\bar{M}(\tau)$, we define $M(\tau)$ as:

$$M(\tau) = \left[m_{p_1}(\tau)...m_{p_n}(\tau)\right]^T \mid n = |P| \qquad (21)$$

where $m_{p_k}(\tau) \mid k = 1,...,n$ is the estimated marking of the place $p_k \in P$. Now, the discrete marking can be obtained with the following procedure.

Algorithm: Defuzzification
See Algorithm A.

Example.
Following the previous example, the marking $M(\tau)$ during $\tau \in (0.08]$ does not change, that is

$$\bar{M}(\tau) = M_{0^+} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Figure 5. (a) Fuzzy marking evolution. (b) Marking estimation. (c) Discrete state
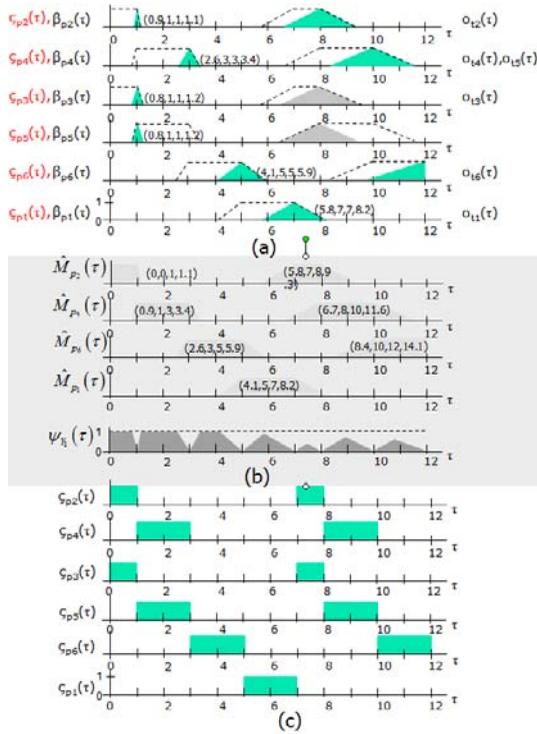
For τ = 0.95 the new fuzzy marking is

$$M\left(0.95\right)=\begin{bmatrix}0 & 1 & 1 & 0.5 & 0.25 & 0\end{bmatrix}^{T},$$

therefore

$$\bar{M}_{1}\left(0\,95\right)=\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}^{T}\quad \bar{M}_{2}\left(0\,95\right)=\begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}^{T}$$

$$\hat{M}\left(0\,95\right)=\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}^{T}+\begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}^{T}=\begin{bmatrix}0 & 1 & 1 & 0 & 0 & 0\end{bmatrix}^{T}$$

$$\bar{M}\left(0\,95\right)=\begin{bmatrix}0 & 1 & 1 & 0 & 0 & 0\end{bmatrix}^{T}$$

Figure 5(c) shows the marking obtained at different instants.

FUTURE TRENDS

Previous results on estimation of Fuzzy Timed State Machines and that included in this article are going to be integrated for addressing a larger class of PN.

Another issue currently addressed is the study of FTPN including measurable places for dealing with sensors or detectable activities within the system; this will allow establishing a bound on the uncertainty of the estimated state. The optimal placement of sensors is an interesting matter of research.

Algorithm A.

> Input: $M\left(\tau\right), Y$
>
> Output: $\bar{M}\left(\tau\right)$
> _____
> Step 1 $\bar{M}\left(\tau\right)\leftarrow\vec{0}$
>
> Step 2 $\forall Y_{i}\mid i=1,...,\left|Y\right|$
>
> > Step 2.1 $\forall p_{k}\in Y_{i}:\ \hat{m}_{q}=\max\left(\bar{M}\left(p_{k}\right)\right)$
> >
> > Step 2.2 $\bar{M}\left(p_{q}\right)=1$

The aim of this research has been the use of the methodology for estimating the DES state of a discrete event system for monitoring its behavior and diagnosing faults. A FTPN is going to be used as a reference model and their outputs (measurable marking) have to be compared with the outputs of the monitored system; the analysis of residuals should provide an early detection of system malfunctioning and a plausible location of the faulty behavior.

CONCLUSION

This article addressed the state estimation problem of DES whose the duration of activities is ill known; fuzzy sets represent the uncertainty of the ending of activities. Several novel notions have been introduced in the FTPN definition, and a new matrix formulation for computing the fuzzy marking of Marked Graphs has been proposed. The extreme situation in which any activity of a system cannot be detected by sensors has been dealt for illustrating the degradation of the marking estimation when a cyclic execution is performed. Current research addresses the topics mentioned in the above section.

REFERENCES

Andreu, D., Pascal, J-C., Valette. R. (1997). Fuzzy Petri Net-Based Programmable Logic Controller. *IEEE Trans. Syst. Man. Cybern*, Vol.27, No. 6, 952-961.

Cao, T., Sanderson, A. C. (1996). Intelligent Task Planning Using Fuzzy Petri Nets. Intelligent Control and Intelligent Automation, Vol. 3. *Word Scientific.*

Cardoso, J., Camargo., H. (1999). Fuzziness in Petri Nets. *Physical Verlag*.

Chen, S., Ke, J., Chang, J. (1990). Knowledge representation using Fuzzy Petri nets. IEEE *Trans. Knowledge Data Eng.*, Vol. 2, No. 3, 311-319.

Ding, Z., Bunke, H., Schneider, M., Kandel., A. (2005). Fuzzy Timed Petri Net, Definitions, Properties, and Applications. *Elsevier, Mathematical and Computer Modelling* 41, 345-360.

Gao, M., Zhou, M., Guang, X., Wu, Z. (2003). Fuzzy Reasoning Petri Nets, *IEEE Trans. Syst., Man &*

Cybern., Part A: Syst. and Humans, Vol. 33, No. 3, 314-324.

Giua, A., Julvez, C., Seatzu C. (2003). Marking Estimation of Petri Nets base on Partial Observation. *Proc. Of the American Control Conference, Denver, Colorado June 4-6,* 326-331.

González-Castolo, J. C., López-Mellado, E. (2006). Fuzzy State Estimation of Discrete Event Systems. *Proc. of MICAI 2006: Advances in Artificial Intelligence,* Vol. 4293, 90-100.

Hennequin, S., Lefebvre, D., El Moudni., A. (2001). Fuzzy Multimodel of Timed Petri Nets. *IEEE Trans. on Syst., Man, Cybern.,* Vol. 31, No. 2, 245-250.

Klir, G. and Yuan, B. (1995). Fuzzy Sets and Fuzzy Logic. Theory and Applications. *Prentice Hall, NJ, USA.*

Koriem, S.M. (2000). A Fuzzy Petri Net Tool For Modeling and Verification of Knowledge-Based Systems. *The Computer Journal*, Vol. 43, No. 3. 206-223.

Leslaw, G., Kluska, J. (2004). Hardware Implementation of Fuzzy Petri Net as a Controller. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 34, No. 3, 1315-1324.

Martinez, J. & Silva, M. (1982). A Simple and fast algorithm to obtain all invariants of a generalized Petri nets. *Proc. of Second European Workshop on Application and Theory of Petri Nets,* Informatik-Fachberichte Vol. 52, 301-310.

Murata., T. (1996). Temporal uncertainty and fuzzy-timing high-level Petri nets. *Lect. Notes Comput. Sci.*, Vol.1091, 29-58.

Pedrycz, W., Camargo, H. (2003). Fuzzy timed Petri Nets. *Elsevier, Fuzzy Sets and Systems,* 140, 301-330.

Ramirez-Treviño, A., Rivera-Rangel, A., López-Mellado, E. (2003). Observability of Discrete Event Systems Modeled by Interpreted Petri Nets. *IEEE Transactions on Robotics and Automation.* Vol.19, No. 4, 557-565.

Shen, R. V. L. (2003). Reinforcement Learning for High-Level Fuzzy Petri Nets. *IEEE Trans. on Syst., Man, & Cybern.,* Vol. 33, No. 2, 351-362.

KEY TERMS

Discrete Events Systems: It is the class of systems whose behavior is characterized by successions of states delimited by asynchronous events. Most of these systems have been man made.

Fuzzy Logic: It is a Knowledge representation technique and computing framework whose approach is based on degrees of truth rather than the usual "true" or "false" of classical logic.

Fuzzy Petri Nets: It is a family of formalisms extending Petri nets by the inclusion of fuzzy sets representing usually uncertainty of time elapses.

Imprecise Marking: The imprecise localization of tokens within places of a FTPN; it is computed as a possibility distribution.

Petri Nets: It is a family of formalisms for modeling and analysis of concurrent DES allowing intuitive graphical descriptions and providing a simple but sound mathematical support. A timed Petri net includes information about the duration of the modeled activities.

Marked Graph: It is a Petri Net subclass in which every place has only one input transition and one output transition.

State Estimation: It is the inference process that determines the current state of a system from the knowledge of sequences of inputs and outputs.

State Machine: It is a Petri Net subclass in which every transition has only one input place and one output place.

System Monitoring: It is a surveillance process on measurable events and/or outputs of a system; it is often used a reference model that specifies a reasonable good behavior. Deviations from the reference are analyzed and determined if there exist a fault. This process is included as a part of a fault diagnosis process.

F

Fuzzy Control Systems: An Introduction

Guanrong Chen
City University of Hong Kong, China

Young Hoon Joo
Kunsan National University, Korea

INTRODUCTION

Fuzzy control systems are developed based on fuzzy set theory, attributed to Lotfi A. Zadeh (Zadeh, 1965, 1973), which extends the classical set theory with memberships of its elements described by the classical characteristic function (either "is" or "is not" a member of the set), to allow for partial membership described by a membership function (both "is" and "is not" a member of the set at the same time, with a certain degree of belonging to the set). Thus, fuzzy set theory has great capabilities and flexibilities in solving many real-world problems which classical set theory does not intend or fails to handle.

Fuzzy set theory was applied to control systems theory and engineering almost immediately after its birth. Advances in modern computer technology continuously backs up the fuzzy framework for coping with engineering systems of a broad spectrum, including many control systems that are too complex or too imprecise to tackle by conventional control theories and techniques.

BACKGROUND: FUZZY CONTROL SYSTEMS

The main signature of fuzzy logic technology is its ability of suggesting an approximate solution to an imprecisely formulated problem. From this point of view, fuzzy logic is closer to human reasoning than the classical logic, where the latter attempts to precisely formulate and exactly solve a mathematical or technical problem if ever possible.

Motivations for Fuzzy Control Systems Theory

Conventional control systems theory, developed based on classical mathematics and the two-valued logic, is relatively mature and complete. This theory has its solid foundation built on classical mathematics, electrical engineering, and computer technology. It can provide rigorous analysis and often perfect solutions when a system is precisely defined mathematically. Within this framework, some relatively advanced control techniques such as adaptive, robust and nonlinear control theories have gained rapid development in the last three decades.

However, conventional control theory is quite limited in modeling and controlling complex dynamical systems, particularly ill-formulated and partially-described physical systems. Fuzzy logic control theory, on the contrary, has shown potential in these kinds of non-traditional applications. Fuzzy logic technology allows the designers to build controllers even when their understanding of the system is still in a vague, incomplete, and developing phase, and such situations are quite common in industrial control practice.

General Structure of Fuzzy Control Systems

Just like other mathematical tools, fuzzy logic, fuzzy set theory, fuzzy modeling, fuzzy control methods, etc., have been developed for solving practical problems. In control systems theory, if the fuzzy interpretation of a real-world problem is correct and if fuzzy theory is developed appropriately, then fuzzy controllers can be suitably designed and they work quite well to their advantages. The entire process is then returned to the

F

original real-world setting, to accomplish the desired system automation. This is the so-called "fuzzification—fuzzy operation—defuzzification" routine in fuzzy control design. The key step—fuzzy operation—is executed by a logical rule base consisting of some IF-THEN rules established by using fuzzy logic and human knowledge (Chen & Pham, 1999, 2006; Drianker, Hellendoorn & Reinfrank, 1993; Passino & Yurkovich, 1998; Tanaka, 1996; Tanaka & wang, 1999; Wang, 1994; Ying, 2000).

Fuzzification

Fuzzy set theory allows partial membership of an element with respect to a set: an element can partially belong to a set and meanwhile partially not belong to the same set. For example, an element, x, belonging to the set, X, IS specified by a (normalized) membership function, $\mu_X : X \to [0,1]$. There are two extreme cases: $\mu_X(x) = 0$ means $x \notin X$ and $\mu_X(x) = 1$ means $x \in X$ in the classical sense. But $\mu_X(x) = 0.2$ means x belongs to X only with grade 0.2, or equivalently, x does not belong to X with grade 0.8. Moreover, an element can have more than one membership value at the same time, such as $\mu_X(x) = 0.2$ and $\mu_X(x) = 0.6$, and they need not be summed up to one. The entire setting depends on how large the set X is (or the sets X and Y are) for the associate members, and what kind of shape a membership function should have in order to make sense of the real problem at hand. A set, X, along with a membership function defined on it, $\mu_X(\cdot)$, is called a *fuzzy set* and is denoted (X, μ_X). More examples of fuzzy sets can be seen below, as the discussion continues. This process of transforming a crisp value of an element (say $x = 0.3$) to a fuzzy set (say $x = 0.3 \in X = [0,1]$ with $\mu_X(x) = 0.2$) is called *fuzzification*.

Given a set of real numbers, $X = [-1,1]$, a point $x \in X$ assumes a real value, say $x = 0.3$. This is a crisp number without fuzziness. However, if a membership function $\mu_X(\cdot)$ is introduced to associate with the set X, then (X, μ_X) becomes a fuzzy set, and the (same) point $x = 0.3$ has a membership grade quantified by $\mu_X(\cdot)$ (for instance, $\mu_X(x) = 0.9$). As a result, x has not one but two values associated with the point: $x = 0.3$ and $\mu_X(x) = 0.9$. In this sense, x is said to have been *fuzzified*. For convenience, instead of saying that "x is in the set X with a membership value $\mu_X(x)$," in common practice it is usually said "x is ," while one should keep in mind that there is always a well-defined membership function

associated with the set X. If a member, x, belongs to two fuzzy sets, one says "x is X_1 AND x is X_2," and so on. Here, the relation AND needs a logical operation to perform. As a result, this statement eventually yields only one membership value for the element x, denoted by $\mu_{X_1 \times X_2}(x)$. There are several logical operations to implement the logical AND; they are quite different but all valid within their individual logical system. A commonly used one is $\mu_{X_1 \times X_2}(x) = \min \{\mu_{X_1}(x), \mu_{X_2}(x)\}$.

Fuzzy Logic Rule Base

The majority of fuzzy logic control systems are knowledge-based systems. This means that either their fuzzy models or their fuzzy logic controllers are described by fuzzy logic IF-THEN rules. These rules have to be established based on human expert's knowledge about the system, the controller, and the performance specifications, etc., and they must be implemented by performing rigorous logical operations.

For example, a car driver knows that if the car moves straight ahead then he does not need to do anything; if the car turns to the right then he needs to steer the car to the left; if the car turns to the right by too much then he needs to take a stronger action to steer the car to the left much more, and so on. Here, "much" and "more" etc. are fuzzy terms that cannot be described by classical mathematics but can be quantified by membership functions (see Fig. 2, where part (a) is an example of the description "to the left"). The collection of all such "if … then …" principles constitutes a *fuzzy logic rule base* for the problem under investigation. To this end, it is helpful to briefly summarize the experience of the driver in the following simplified rule base: Let $X = [-180°, 180°]$, x be the position of the car, $\mu_{left}(\cdot)$ be the membership function for the moving car turning "to the left," $\mu_{right}(\cdot)$ the membership function for the car turning "to the right," and $\mu_0(\cdot)$ the membership function for the car "moving straight ahead." Here, simplified statements are used, for instance, "x is X_{left}" means "x belongs to X with a membership value $\mu_{left}(x)$" etc. Also, similar notation for the control action u of the driver is employed. Then, a simple typical rule base for this car-driving task is

$R^{(1)}$:   IF x is X_{left} THEN u is U_{right}
$R^{(2)}$:   IF x is X_{right} THEN u is U_{left}
$R^{(3)}$:   IF x is X_0 THEN u is U_0

where X_0 means moving straight ahead (not left nor right), as described by the membership function shown in Fig. 2(c), and "u is U_0" means $u = 0$ (no control action) with a certain grade (if this grade is 1, it means absolutely no control action). Of course, this description only illustrates the basic idea, which is by no means a complete and effective design for a real car-driving application.

In general, a rule base of r rules has the form

$$R^{(k)}: \text{IF } x_1 \text{ is } X_{k1} \text{ AND } \cdots \text{ AND } x_m \text{ is } X_{km} \text{ THEN } u \text{ is } U_k \quad (1)$$

where $m \geq 1$ and $k = 1, \cdots, r$.

Defuzzification

An element of a fuzzy set may have more than one membership value. In Fig. 1, for instance, if $x = 5°$ then it has two membership values: $\mu_{right}(x) = 5/180 \approx 0.28$ and $\mu_0(x) = 0.5$. This means that the car is moving to the right by a little. According to the above-specified rule base, the driver will take two control actions simultaneously, which is unnecessary and physically impossible.

Thus, what should the control action be? To simplify this discussion, suppose that the control action is simply $u = -x$ with the same membership functions $\mu_X = \mu_U$ for all cases. Then, a natural and realistic control action for the driver to take is a compromise between the two required actions. Among several possible compromise (or, average) formulas for this purpose, the most commonly adopted one that works well in most cases is the following weighted average formula:

$$u = \frac{\mu_{right}(u) \cdot u + \mu_0(u) \cdot u}{\mu_{right}(u) + \mu_0(u)} =$$

$$\frac{0.28 \times (-5°) + 0.5 \times (-5°)}{0.28 + 0.5} = -0.5°$$

Here, the result is interpreted as "the driver should turn the car to the left by 5°." This averaged outputs is called *defuzzification*, which yields a single crisp value for the control, which may actually yield similar averaged results in general.

The result of defuzzification usually is a physical quantity acceptable by the original real system. Whether or not this defuzzification result works well depends

Figure 1. Membership functions for directions of a moving car

(a) Membership function for "to the left."

(b) Membership function for "to the right."

(c) Membership function for "straight ahead."

F
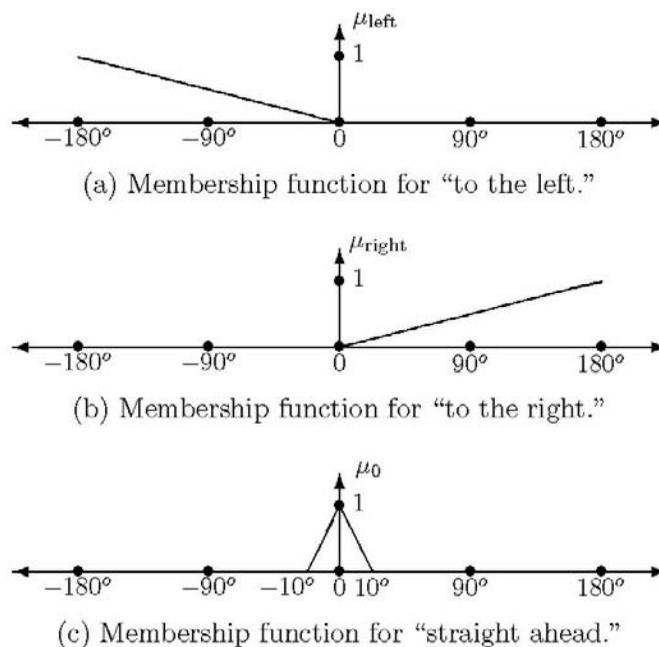
Figure 2. A typical fuzzy logic controller
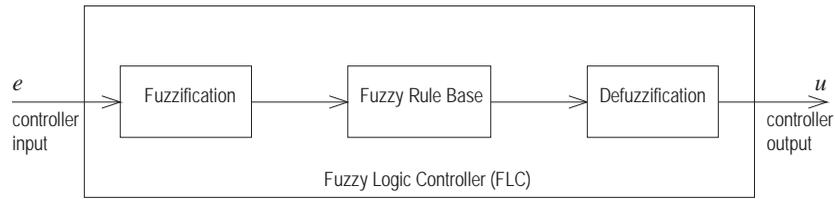
on the correctness and effectiveness of the rule base, while the latter depends on the designer's knowledge and experience about the physical system or process for control. Just like any of the classical design problems, there is generally no unique solution for a problem; an experienced designer usually comes out with a better design.

A general weighted average formula for defuzzification is the following convex combination of the individual outputs:

$$output = \sum_{i=1}^{r} \alpha_i u_i := \sum_{i=1}^{r} \frac{w_i}{\sum_{i=1}^{r} w_i} \cdot u_i \qquad (2)$$

with notation referred to the rule base (1), where

$$w_i \mu_{U_i}(u_i), \quad \alpha_i := \frac{w_i}{\sum_{i=1}^{r} w_i} \geq 0, \quad i = 1, \cdots, r, \quad \sum_{i=1}^{r} \alpha_i = 1$$

Sometimes, depending on the design or application, the weights are

$$w_i = \prod_{j=1}^{m} \mu_{X_{ij}}(x_j), \quad i = 1, \cdots, r$$

The overall structure of a fuzzy logic controller is shown in Fig. 2.

MAIN FOCUS OF THE CHAPTER: SOME BASIC FUZZY CONTROL APPROACHES

A Model-Free Approach

This general approach of fuzzy logic control works for trajectory tracking for a conventional dynamical system that does not have a precise mathematical model.

The basic setup is shown in Fig. 3, where the plant is a conventional system without a mathematical description and all the signals (the reference set-point *sp*, output $y(t)$, control $u(t)$, and error $e(t) = sp - y(t)$) are crisp. The objective is to design a controller to achieve the goal $e(t) \rightarrow 0$ as $t \rightarrow \infty$, assuming that the system inputs and outputs are measurable by sensors on line.

If the mathematical formulation of the plant is unknown, how can one develop a controller to control this plant? Fuzzy logic approach turns out to be advantageous in this situation: it only uses the plant inputs and outputs, but not the state variables nor any other information. After the design is completed, the entire dashed-block in Fig. 2 is used to replace the "controller" block in Fig. 3.

As an example, suppose that the physical reference set-point is the degree of temperature, say $40°F$, and that the designer knows the range of the error signal, $e(t) = 40° - y(t)$, is within $X = [-25°, 45°]$, and assume that the scale of control is required to be in the unit of $1°$. Then, the membership functions for the error signal to be "negative large" (NL), "positive large" (PL), and "zero" (ZO) may be chosen as shown in Fig.

Figure 3. A typical reference set-point tracking control system
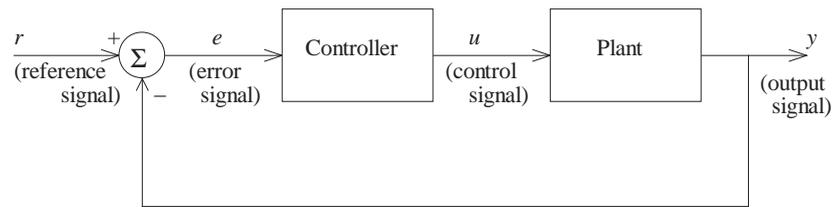
4. Using these membership functions, the controller is expected to drive the output temperature to be within the allowable range: $40° \pm 1°$. With these membership functions, when the error signal $e(t) = 5°$, for instance,

Figure 4. Membership function for the error temperature signal

it is considered to be "positive large" with membership value one, meaning that the set-point ($40°$) is higher than $y(t)$ by too much.

The output from the fuzzification module is a fuzzy set consisting of the interval X and three membership functions, μ_{NL}, μ_{PL} and μ_{ZO}, in this example. The output from fuzzification will be the input to the next module—the fuzzy logic rule base—which only takes fuzzy set inputs to be compatible with the logical IF-THEN rules.

Figure 5 is helpful for establishing the rule base. If $e > 0$ at a moment, then the set-point is higher than the output y (since $e = 40° - y$), which corresponds to two possible situations, marked by a and d respectively. To further distinguish these two situations, one may use the rate of change of the error, $\dot{e} = -\dot{y}$. Here, since the set-point is a constant, its derivative is zero. Using information from both e and $\dot{e}$, one can completely characterize the changing situation of the output temperature at all times. If, for example, $e > 0$ and $\dot{e} > 0$,

Figure 5. Temperature set-point tracking example
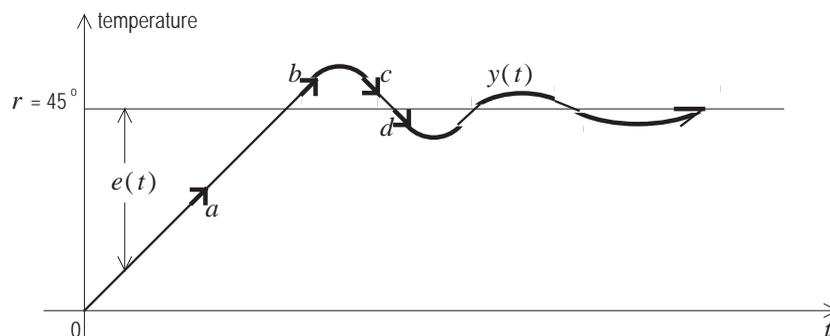
then the temperature is currently at situation *d* rather than situation *a*, since $\dot{e}>0$ means $\dot{y}<0$ which, in turn, signifies that the curve is moving downward.

Based on the above observation from the physical situations of the current temperature against the set-point, a simple rule base can be established as follows:

R^1:　IF $e>0$ AND >0 THEN $u(t+)=-C\cdot u(t)$;
R^2:　IF $e>0$ AND <0 THEN $u(t+)=C\cdot u(t)$;
R^3:　IF $e<0$ AND >0 THEN $u(t+)=C\cdot u(t)$;
R^4:　IF $e<0$ AND <0 THEN $u(t+)=-C\cdot u(t)$;

otherwise (e.g., $e=0$ or $\dot{e}=0$), $u(t+)=u(t)$, till next step, where $C>0$ is a constant control gain and $t+$ can be just $t+1$ in discrete time.

In the above, the first two rules are understood as follows (other rules can be similarly interpreted):

1.　$R^{(1)}$: $e>0$ and >0. As analyzed above, the temperature curve is currently at situation *d*, so the controller has to change its moving direction to the opposite by changing the current control action to the opposite (since the current control action is driving the output curve downward).
2.　$R^{(2)}$: $e>0$ and <0. The current temperature curve is at situation *a*, so the controller does not need to do anything (since the current control action is driving the output curve up toward the set-point).

The switching control actions may take different forms, depending on the design. One example is $u(t+1)=u(t)+\Delta u(t)$, among others (Chen & Pham, 1999, 2006).

Furthermore, to distinguish "positive large" from just "positive" for $e>0$, one may use those membership functions shown in Fig. 4. Since the error signal $e(t)$ is fuzzified in the fuzzification module, one can similarly fuzzify the auxiliary signal $\dot{e}(t)$ in the fuzzification module. Thus, there are two fuzzified inputs, e and $\dot{e}$, for the controller, and they both have corresponding membership functions describing their properties as "positive large" (μ_{PL}), "negative large" (μ_{NL}), or "zero" (μ_{ZO}), as shown in Fig. 5. Thus, for the rule base, one may replace it by a set of more detailed rules as follows:

R^1:　IF $e=$ PL AND >0 THEN $u(t+1)=-\mu_{PL}(e)\cdot u(t)$;
R^2:　IF $e=$ PS AND >0 THEN $u(t+1)=-(1-\mu_{PS}(e))\cdot u(t)$;
R^3:　IF $e=$ PL AND <0 THEN $u(t+1)=\mu_{PL}(e)\cdot u(t)$;
R^4:　IF $e=$ PS AND <0 THEN $u(t+1)=(1-\mu_{PS}(e))\cdot u(t)$;
R^5:　IF $e=$ NL AND >0 THEN $u(t+1)=\mu_{NL}(e)\cdot u(t)$;
R^6:　IF $e=$ NS AND >0 THEN $u(t+1)=(1-\mu_{NS}(e))\cdot u(t)$;
R^7:　IF $e=$ NL AND <0 THEN $u(t+1)=-\mu_{NL}(e)\cdot u(t)$;
R^8:　IF $e=$ NS AND <0 THEN $u(t+1)=-(1-\mu_{NS}(e))\cdot u(t)$;

otherwise, $u(t+1)=u(t)$. Here and below, "= PL" means "is PL," etc. In this way, the initial rule base is enhanced and extended.

In the defuzzification module, new membership functions are needed for the change of the control action, $u(t+1)$ or $\Delta u(t)$, if the enhanced rule base described above is used. This is because both the error and the rate of change of the error signals have been fuzzified to be "positive large" or "positive small," the control actions have to be fuzzified accordingly (to be "large" or "small").

Now, suppose that a complete, enhanced fuzzy logic rule base has been established. Then, in the defuzzification module, the weighted average formula can be used to obtain a single crisp value as the control action output from the controller (see Fig. 2):

$$u(t+1)=\frac{\sum_{i=1}^{N}\mu_i\cdot u_i(t+1)}{\sum_{i=1}^{N}\mu_i}$$

This is an average value of the multiple ($N=8$ in the above rule base) control signals at step $t+1$, and is physically meaningful to the given plant.

A Model-Based Approach

If a mathematical model of the system, or a fairly good approximation of it, is available, one may be able to design a fuzzy logic controller with better results such as performance specifications and guaranteed stability.

F

This constitutes a model-based fuzzy control approach (Chen & Zhang, 1997; Malki, Li & Chen, 1994; Malki, Feigenspan, Misir & Chen, 1997; Sooraksa & Chen, 1998; Ying, Siler & Buckley, 1990).

For instance, a locally linear fuzzy system model is described by a rule base of the following form:

$$R_S^{(k)} : IF \ x_1 \ is \ X_{k1} \quad AND \ \cdots$$
$$AND \quad x_m \ is \ X_{km} \quad THEN \ \underline{\dot{x}} = A_k \, \underline{x} + B_k \, \underline{u} \tag{3}$$

where $\{A_k\}$ and $\{B_k\}$ are given constant matrices, $\underline{x} = [x_1,...,x_m]^T$ is the state vector, and $\underline{u} = [u_1,...,u_n]^T$ is a controller to be designed, with $m \geq n \geq 1$, and $k = 1,\cdots,r$. The fuzzy system model (3) may be rewritten in a more compact form as follows:

$$\underline{\dot{x}} = \sum_{k=1}^{r} \alpha_k \left(A_k \, \underline{x} + B_k \, \underline{u} \right) = A(\mu(\underline{x}))\underline{x} + B(\mu(\underline{x}))\underline{u} \tag{4}$$

where

$$\mu(\underline{x}) = \left\{ \mu_{X_{ij}}(\underline{x}) \right\}_{i,j=1}^{m} .$$

Based on this fuzzy model, (3) or (4), a fuzzy controller $\underline{u}(t)$ can be designed by using some conventional techniques. For example, if a negative state-feedback controller is preferred, then one may design a controller described by the following ruse base:

$$R_C^{(k)} : IF \ x_1 \ is \ X_{k1} \quad AND \ \cdots$$
$$AND \quad x_m \ is \ X_{km} \quad THEN \ \underline{u} = -K_k \, \underline{x} \tag{5}$$

where $\{K_k\}_{k=1}^{r}$ are constant control gain matrices to be determined, $k = 1,\cdots,r$. Thus, the closed-loop controlled system (4) together with (5) becomes

$$R_{SC}^{(k)} : IF \ x_1 \ is \ X_{k1} \quad AND \ \cdots$$
$$AND \quad x_m \ is \ X_{km} \quad THEN \ \underline{\dot{x}} = [A_k - BK_k]\underline{x} \tag{6}$$

For this feedback controlled system, the following is a typical stability condition [1,2,10]:

If there exists a common positive definite and symmetric constant matrix P such that $A_k^T P + PA_k = -Q$ for some

$Q > 0$ for all $k = 1,\cdots,r$, then the fuzzy controlled system (6) is asymptotically stable about zero.

This theorem provides a basic (sufficient) condition for the global asymptotic stability of the fuzzy control system, which can also be viewed as a criterion for tracking control of the system trajectory to the zero set-point. Clearly, stable control gain matrices $\{K_k\}_{k=1}^{r}$ may be determined according to this criterion in a design.

FUTURE TRENDS

This topic will be discussed elsewhere in the near future.

CONCLUSION

The essence of systems control is to achieve automation. For this purpose, a combination of fuzzy control technology and advanced computer facility available in the industry provides a promising approach that can mimic human thinking and linguistic control ability, so as to equip the control systems with a certain degree of artificial intelligence. It has now been realized that fuzzy control systems theory offers a simple, realistic and successful addition, or sometimes an alternative, for controlling various complex, imperfectly modeled, and highly uncertain engineering systems, with a great potential in many real-world applications.

REFERENCES

G. Chen & T. T. Pham (1999). *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. CRC Press.

G. Chen & T. T. Pham (2006) *Introduction to Fuzzy Systems*. CRC Press.

G. Chen & D. Zhang (1997). Back-driving a truck with suboptimal distance trajectories: A fuzzy logic control approach. *IEEE Trans. on Fuzzy Systems*, 5: 369-380.

D. Drianker, H. Hellendoorn & M. Reinfrank (1993). *An Introduction to Fuzzy Control*. Springer-Verlag.

H. Malki, D. Feigenspan, D. Misir & G. Chen (1997) Fuzzy PID control of a flexible-joint robot arm with uncertainties from time-varying loads. *IEEE Trans. on Contr. Sys. Tech*. 5: 371-378.

H. Malki, H. Li & G. Chen (1994). New design and stability analysis of fuzzy proportional-derivative control systems. *IEEE Trans. on Fuzzy Systems*, 2: 345-354.

K. M. Passino & S. Yurkovich (1998) *Fuzzy Control*, Addison-Wesley.

P. Sooraksa & G. Chen (1998). Mathematical modeling and fuzzy control of flexible robot arms. *Math. Comput. Modelling*. 27: 73-93.

K. Tanaka (1996). *An Introduction to Fuzzy Logic for Practical Applications*. Springer.

K. Tanaka & H. O. Wang (1999). *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. IEEE Press.

L. X. Wang (1994) *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Prentice-Hall.

H. Ying (2000). *Fuzzy Control and Modeling: Analytical Foundations and Applications*. IEEE Press.

H. Ying, W. Siler & J. J. Buckley (1990). Fuzzy control theory: a nonlinear case. *Automatica*. 26: 513-520.

L. A. Zadeh (1965). Fuzzy sets. *Information and Control*. 8: 338-353.

L. A. Zadeh (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. on Systems, Man, and Cybernetics*. 3: 28-44.

F

KEY TERMS

Defuzzification: A process that converts fuzzy terms to conventional expressions quantified by real-valued functions.

Fuzzification: A process that converts conventional expressions to fuzzy terms quantified by fuzzy membership functions.

Fuzzy Control: A control method based on fuzzy set and fuzzy logic theories.

Fuzzy Logic: A logic that takes on continuous values in between 0 and 1.

Fuzzy Membership Function: A function defined on fuzzy set and assumes continuous values in between 0 and 1.

Fuzzy Set: A set of elements with a real-valued membership function describing their grades.

Fuzzy System: A system formulated and described by fuzzy set-based real-valued functions.

Fuzzy Decision Trees

Malcolm J. Beynon
Cardiff University, UK

INTRODUCTION

The inductive learning methodology known as decision trees, concerns the ability to classify objects based on their attributes values, using a tree like structure from which decision rules can be accrued. In this article, a description of decision trees is given, with the main emphasis on their operation in a fuzzy environment.

A first reference to decision trees is made in Hunt *et al.* (1966), who proposed the Concept learning system to construct a decision tree that attempts to minimize the score of classifying chess endgames. The example problem concerning chess offers early evidence supporting the view that decision trees are closely associated with artificial intelligence (AI). It is over ten years later that Quinlan (1979) developed the early work on decision trees, to introduced the Interactive Dichotomizer 3 (ID3). The important feature with their development was the use of an entropy measure to aid the decision tree construction process (using again the chess game as the considered problem).

It is ID3, and techniques like it, that defines the hierarchical structure commonly associated with decision trees, see for example the recent theoretical and application studies of Pal and Chakraborty (2001), Bhatt and Gopal (2005) and Armand *et al.* (2007). Moreover, starting from an identified root node, paths are constructed down to leaf nodes, where the attributes associated with the intermediate nodes are identified through the use of an entropy measure to preferentially gauge the classification certainty down that path. Each path down to a leaf node forms an 'if .. then ..' decision rule used to classify the objects.

The introduction of fuzzy set theory in Zadeh (1965), offered a general methodology that allows notions of vagueness and imprecision to be considered. Moreover, Zadeh's work allowed the possibility for previously defined techniques to be considered with a fuzzy environment. It was over ten years later that the area of decision trees benefited from this fuzzy environment opportunity (see Chang and Pavlidis, 1977). Since then there has been a steady stream of research studies that have developed or applied fuzzy decision trees (FDTs) (see recently for example Li *et al.*, 2006 and Wang *et al.*, 2007).

The expectations that come with the utilisation of FDTs are succinctly stated by Li *et al.* (2006, p. 655);

"Decision trees based on fuzzy set theory combines the advantages of good comprehensibility of decision trees and the ability of fuzzy representation to deal with inexact and uncertain information."

Chiang and Hsu (2002) highlight that decision trees has been successfully applied to problems in artificial intelligence, pattern recognition and statistics. They go onto outline a positive development the FDTs offer, namely that it is better placed to have an estimate of the degree that an object is associated with each class, often desirable in areas like medical diagnosis (see Quinlan (1987) for the alternative view with respect to crisp decision trees).

The remains of this article look in more details at FDTs, including a tutorial example showing the rudiments of how an FDT can be constructed.

BACKGROUND

The background section of this article concentrates on a brief description of fuzzy set theory pertinent to FDTs, followed by a presentation of one FDT technique.

In fuzzy set theory (Zadeh, 1965), the grade of membership of a value x to a set S is defined through a membership function $\mu_S(x)$ that can take a value in the range [0, 1]. The accompanying numerical attribute domain can be described by a finite series of MFs that each offers a grade of membership to describe x, which collectively form its concomitant fuzzy number. In this article, MFs are used to formulate linguistic variables for the considered attributes. These linguistic variables are made up of sets of linguistic terms which are defined by the MFs (see later).

Figure 1. Example membership function and their use in a linguistic variable
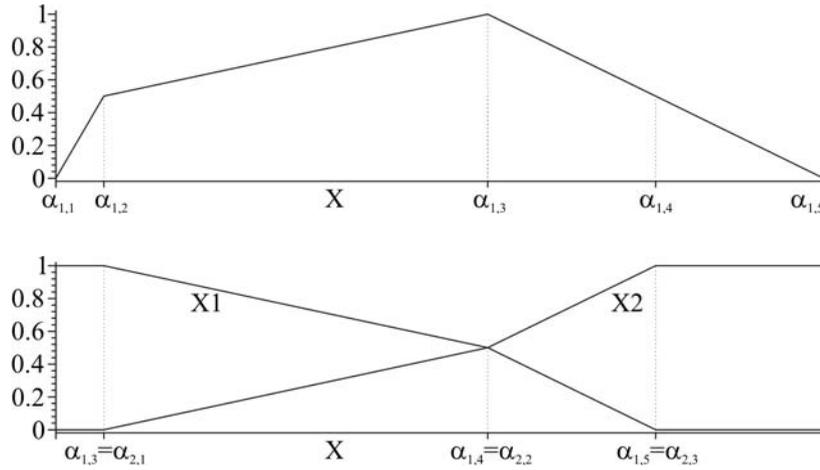
Surrounding the notion of MFs is the issue of their structure (Dombi and Gera, 2005). Here, piecewise linear MFs are used to define the linguistic terms presented, see Figure 1.

In Figure 1(top), a single piecewise linear MF is shown along with the defining values that define it, namely, $\alpha_{1,1}$, $\alpha_{1,2}$, $\alpha_{1,3}$, $\alpha_{1,4}$ and $\alpha_{1,5}$. The associated mathematical structure of this specific form of MF is given below;

$$\mu(x) \begin{cases} 0 & \text{if } x \le \alpha_{j,1} \\ 0.5\dfrac{x-\alpha_{j,1}}{\alpha_{j,2}-\alpha_{j,1}} & \text{if } \alpha_{j,1} < x \le \alpha_{j,2} \\ 0.5+0.5\dfrac{x-\alpha_{j,2}}{\alpha_{j,3}-\alpha_{j,2}} & \text{if } \alpha_{j,2} < x \le \alpha_{j,3} \\ 1 & \text{if } x = \alpha_{j,3} \\ 1-0.5\dfrac{x-\alpha_{j,3}}{\alpha_{j,4}-\alpha_{j,3}} & \text{if } \alpha_{j,3} < x \le \alpha_{j,4} \\ 0.5-0.5\dfrac{x-\alpha_{j,4}}{\alpha_{j,5}-\alpha_{j,4}} & \text{if } \alpha_{j,4} < x \le \alpha_{j,5} \\ 0 & \text{if } \alpha_{j,5} < x \end{cases}$$

As mentioned earlier, MFs of this type are used to define the linguistic terms which make up linguistic variables. An example of a linguistic variable X based on two linguistic terms, X1 and X2, is shown in Figure 2(bottom), where the overlap of the defining values for

each linguistic term is evident. Moreover, using left and right limits of the X domain as $-\infty$ and ∞, respectively, the sets of defining values are (in list form); X1 - $[-\infty, -\infty, \alpha_{1,3}, \alpha_{1,4}, \alpha_{1,5}]$ and X2 - $[\alpha_{2,1}, \alpha_{2,2}, \alpha_{2,3}, \infty, \infty]$, where $\alpha_{1,3} = \alpha_{2,1}$, $\alpha_{1,4} = \alpha_{2,2}$ and $\alpha_{1,5} = \alpha_{2,3}$.

This section now goes on to outline the technical details of the fuzzy decision tree approach introduced in Yuan and Shaw (1995). With an inductive fuzzy decision tree, the underlying knowledge related to a decision outcome can be represented as a set of fuzzy 'if .. then ..' decision rules, each of the form;

If $(A_1$ is $T_{i_1}^1)$ and $(A_2$ is $T_{i_2}^2)$ … and $(A_k$ is $T_{i_k}^k)$ then C is C_j, where $A_1, A_2, .., A_k$ and C are linguistic variables for the multiple antecedents $(A_i$'s) and consequent (C) statements used to describe the considered objects, and $T(A_k) = \{T_1^k, T_2^k, .. T_{S_i}^k\}$ and $\{C_1, C_2, …, C_L\}$ are their respective linguistic terms, defined by the MFs $\mu_{T_j^k}(x)$ etc. The MFs, $\mu_{T_j^k}(x)$ and $\mu_{C_j}(y)$, represent the grade of membership of an object's antecedent A_j being T_j^k and consequent C being C_j, respectively.

A MF $\mu(x)$ from the set describing a fuzzy linguistic variable Y defined on X, can be viewed as a possibility distribution of Y on X, that is $\pi(x) = \mu(x)$, for all $x \in X$ the values taken by the objects in U (also normalized so $\max_{x \in X} p(x) = 1$). The possibility measure $E_\alpha(Y)$ of ambiguity is defined by

$$E_\alpha(Y) = g(\pi) = \sum_{i=1}^{n}(p_i^* - p_{i+1}^*)\ln[i],$$

where $\pi^* = \{\pi_1{}^*, \pi_2{}^*, \ldots, \pi_n{}^*\}$ is the permutation of the normalized possibility distribution $\pi = \{\pi(x_1), \pi(x_2), \ldots, \pi(x_n)\}$, sorted so that $\pi_i{}^* \geq \pi_{i+1}^*$ for $i = 1, .., n$, and $p_{n+1}^* = 0$.

The ambiguity of attribute A (over the objects $u_1, .., u_m$) is given as:

$$E_\alpha(A) = \frac{1}{m}\sum_{i=1}^{m} E_a\ (A(u_i)\ ,$$

where

$$E_\alpha(A(u_i)) = g(\mu_{Ts}(u_i)/\max_{1\leq j\leq s}(\mu_{Tj}(u_i))),$$

with $T_1, \ldots, T_s$ the linguistic terms of an attribute (antecedent) with m objects.

The fuzzy subsethood $S(A, B)$ measures the degree to which A is a subset of B, and is given by,

$$S(A, B) = \sum_{u\in U}\min(\mu_A(u), \mu_B(u))\bigg/\sum_{u\in U}\mu_A(u).$$

Given fuzzy evidence E, the possibility of classifying an object to the consequent C_i can be defined as,

$$\pi(C_i/E) = S(E, C_i)/\max_j S(E, C_j),$$

where the fuzzy subsethood $S(E, C_i)$ represents the degree of truth for the classification rule ('if E then C_i'). With a single piece of evidence (a fuzzy number for an attribute), then the classification ambiguity based on this fuzzy evidence is defined as: $G(E) = g(\pi(C|E))$, which is measured using the possibility distribution $\pi(C|E) = (\pi(C_1|E), \ldots, \pi(C_L|E))$.

The classification ambiguity with fuzzy partitioning $P = \{E_1, \ldots, E_k\}$ on the fuzzy evidence F, denoted as $G(P|F)$, is the weighted average of classification ambiguity with each subset of partition:

$$G(P|F) = \sum_{i=1}^{k} w(E_i|F)G(E_i\cap F),$$

where $G(E_i\cap F)$ is the classification ambiguity with fuzzy evidence $E_i\cap F$, and where $w(E_i|F)$ is the weight which represents the relative size of subset $E_i\cap F$ in

F: $w(E_i|F) =$

$$\sum_{u\in U}\min(\mu_{E_i}(u), \mu_F(u))\bigg/\sum_{j=1}^{k}\bigg(\sum_{u\in U}\min(\mu_{E_j}(u), \mu_F(u))\bigg)$$

In summary, attributes are assigned to nodes based on the lowest level of classification ambiguity. A node becomes a leaf node if the level of subsethood is higher than some truth value β assigned to the whole of the fuzzy decision tree. The classification from the leaf node is to the decision group with the largest subsethood value. The truth level threshold β controls the growth of the tree; lower β may lead to a smaller tree (with lower classification accuracy), higher β may lead to a larger tree (with higher classification accuracy).

MAIN THRUST

The main thrust of this article is a detailed example of the construction of a fuzzy decision tree. The description includes the transformation of a small data set into a fuzzy data set where the original values are described by their degrees of membership to certain linguistic terms.

The small data set considered, consists of five objects, described by three condition attributes T1, T2 and T3, and classified by a single decision attribute C, see Table 1.

If these values are considered imprecise, fuzzy, there is the option to transform the data values in

Table 1. Example data set

Object	T1	T2	T3	C
u_1	112	45	205	7
u_2	85	42	192	17
u_3	130	58	188	22
u_4	93	54	203	29
u_5	132	39	189	39

Figure 2. Membership functions defining the linguistic terms, C_L and C_H, for the decision attribute C
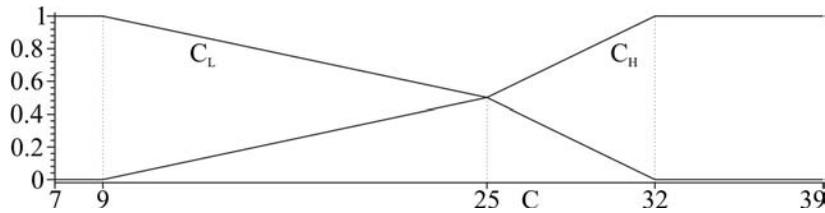
fuzzy values. Here, an attribute is transformed into a linguistic variable, each described by two linguistic terms, see Figure 2.

In Figure 2, the decision attribute C is shown to be described by the linguistic terms, C_L and C_H (possibly denoting the terms low and high). These linguistic terms are themselves defined by MFs ($\mu_{C_L}(\cdot)$ and $\mu_{C_H}(\cdot)$. The hypothetical MFs shown have the respective defining terms of, $\mu_{C_L}(\cdot)$: $[-\infty, -\infty, 9, 25, 32]$ and $\mu_{C_H}(\cdot)$: $[9, 25,$ $32, \infty, \infty]$. To demonstrate their utilisation, for the object u_2, with a value $C = 17$, its fuzzification creates the two values $\mu_{C_L}(17) = 0.750$ and $\mu_{C_H}(17) = 0.250$, the larger of which is associated with the high linguistic term.

A similar series of membership functions can be constructed for the three condition attributes, T1, T2 and T3, Figure 3.

In Figure 3, the linguistic variable version of each condition attribute is described by two linguistic terms

Figure 3. Membership functions defining the linguistic terms for the condition attributes, T1, T2 and T3
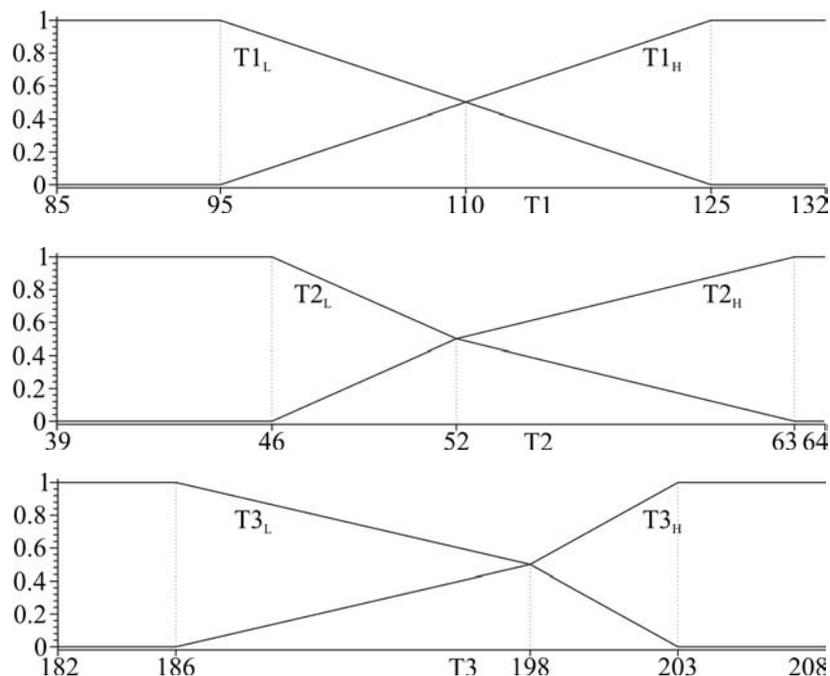
Table 2. Fuzzified version of the example data set

Object	T1 = [T1$_L$, T1$_H$]	T2 = [T2$_L$, T2$_H$]	T3 = [T3$_L$, T3$_H$]	C = [C$_L$, C$_H$]
o_1	[0.433, **0.567**] - H	[**1.000**, 0.000] - L	[0.000, **1.000**] - H	[**1.000**, 0.000] - L
o_2	[**1.000**, 0.000] - L	[**1.000**, 0.000] - L	[**0.750**, 0.250] - L	[**0.750**, 0.250] - L
o_3	[0.000, **1.000**] - H	[0.227, **0.773**] - H	[**0.917**, 0.083] - L	[**0.594**, 0.406] - L
o_4	[**1.000**, 0.000] - L	[0.409, **0.591**] - H	[0.000, **1.000**] - H	[0.214, **0.786**] - H
o_5	[0.000, **1.000**] - H	[**1.000**, 0.000] - L	[**0.875**, 0.125] - L	[0.000, **1.000**] - H

(possibly termed as low and high), themselves defined by MFs. The use of these series of MFs is the ability to fuzzify the example data set, see Table 2.

In Table 2, each object is described by a series of fuzzy values, two fuzzy values for each attribute. Also shown in Table 2, in bold, are the larger of the values in each pair of fuzzy values, with the respective linguistic term this larger value is associated with. Beyond the fuzzification of the data set, attention turns to the construction of the concomitant fuzzy decision tree for this data. Prior to this construction process, a threshold value of β = 0.75 for the minimum required truth level was used throughout.

The construction process starts with the condition attribute that is the root node. For this, it is necessary to calculate the classification ambiguity $G(E)$ of each condition attribute. The evaluation of a $G(E)$ value is shown for the first attribute T1 (i.e. $g(\pi(C|\text{T1}))$), where it is broken down to the fuzzy labels L and H, for L;

$$\pi(C/\text{T1}_L) = S(\text{T1}_L, C_i) / \max_j S(\text{T1}_L, C_j),$$

considering C_L and C_H with the information in Table 1;

$$S(\text{T1}_L, C_L) = \sum_{u \in U} \min(\mu_{\text{T1}_L}(u), \mu_{C_L}(u)) \Big/ \sum_{u \in U} \mu_{\text{T1}_L}(u) =$$

$$1.398/2.433 = 0.574,$$

whereas, $S(\text{T1}_L, C_H) = 0.426$. Hence $\pi = \{0.574, 0.426\}$, giving the ordered normalized form of $\pi^* = \{1.000, 0.741\}$, with $p_3^* = 0$, then

$$G(\text{T1}_L) = g(\pi(C|\text{T1}_L)) = \sum_{i=1}^{2} (\pi_i^* - \pi_{i+1}^*) \ln[i] = 0.514,$$

along with $G(\text{T1}_H) = 0.572$, then $G(\text{T1}) = (0.514 + 0.572)/2 = 0.543$. Compared with $G(\text{T2}) = 0.579$ and $G(\text{T2}) = 0.583$, the condition attribute T1, with the least classification ambiguity, forms the root node for the desired fuzzy decision tree.

The subsethood values in this case are; for T1: $S(\text{T1}_L, C_L) = \textbf{0.574}$ and $S(\text{T1}_L, C_H) = \textbf{0.426}$, and $S(\text{T2}_H, C_L) = 0.452$ and $S(\text{T2}_H, C_H) = \textbf{0.548}$. For T2$_L$ and T2$_H$, the larger subsethood value (in bold), defines the possible classification for that path. In both cases these values are less that the threshold truth value 0.75 employed, so neither of these paths can be terminated to a leaf node, instead further augmentation of them is considered.

With three condition attributes included in the example data set, the possible augmentation to T1$_L$ is with either T2 or T3. Concentrating on T2, where with $G(\text{T1}_L) = 0.\,0.514$, the ambiguity with partition evaluated for T2 ($G(\text{T1}_L \text{ and } \text{T2}|C)$) has to be less than this value, where;

$$G(\text{T1}_L \text{ and } \text{T2}|C) = \sum_{i=1}^{k} w(\text{T2}_i | \text{T1}_L) G(\text{T1}_L \cap \text{T2}_i).$$

Starting with the weight values, in the case of T1$_L$ and T2$_L$, it follows;

$$w(\text{T2}_L | \text{T1}_L) =$$

$$\sum_{u \in U} \min(\mu_{\text{T2}_L}(u), \mu_{\text{T1}_L}(u)) \Big/ \sum_{j=1}^{k} \left(\sum_{u \in U} \min(\mu_{\text{T2}_j}(u), \mu_{\text{T1}_L}(u)) \right) =$$

$$1.842/2.433 = 0.757.$$

Similarly $w(\text{T2}_H | \text{T1}_L) = 0.243$, hence;

$G(\text{T1}_L \text{ and T2}| C) = 0.757 \times G(\text{T1}_L \cap \text{T2}_L) + 0.699 \times G(\text{T1}_L \cap \text{T2}_H) = 0.757 \times 0.327 + 0.699 \times 0.251 = 0.309$,

A concomitant value for $G(\text{T1}_L \text{ and T3}| C) = 0.487$, the lower of these $(G(\text{T1}_L \text{ and T2}| C))$ is lower than the concomitant $G(\text{T1}_L) = 0.514$, so less ambiguity would be found if the T2 attribute was augmented to the path T1 = L. The subsequent subsethood values in this case for each new path are; T2_L; $S(\text{T1}_L \cap \text{T2}_L, C_L) = \mathbf{0.759}$ and $S(\text{T1}_L \cap \text{T2}_L, C_H) = \mathbf{0.358}$; T2_H: $S(\text{T1}_L \cap \text{T2}_H, C_L) = 0.363$ and $S(\text{T1}_L \cap \text{T2}_H, C_H) = \mathbf{1.000}$. With each suggested classification path, the largest subsethood value is above the truth level threshold, therefore they are both leaf nodes leading from the T1 = L path. The construction process continues in a similar vein for the path T1 = H, with the resultant fuzzy decision tree in this case presented in Figure 4.

The fuzzy decision tree in Figure 8 shows five rules (leaf nodes), **R1**, **R2**, …, **R5**, have been constructed. There are a maximum of four levels to the tree shown, indicating a maximum of three condition attributes are used in the rules constructed. In each non-root node shown the subsethood levels to the decision attribute terms C = L and C = H are shown. On the occasions when the larger of the subsethood values is above the

defined threshold value of 0.75 then they are shown in bold and accompany the node becoming a leaf node.

The interpretative power of FDTs is shown by consideration of the rules constructed. For the rule R5 it can be written down as;

'If T1 = H and T3 = H then C = L with truth level 0.839.'

The rules can be considered in a more linguistic form, namely;

'If T1 is low and T3 is high then C is low with truth level 0.839.'

It is the rules like this one shown that allow the clearest interpretability to the understanding of results in classification problems when using FDTs.

FUTURE TRENDS

Fuzzy decision trees (FDTs) benefit from the inductive learning approach that underpins their construction, to aid in the classification of objects based on their values over different attribute. Their construction in a fuzzy

Figure 4. Fuzzy decision tree for example data set
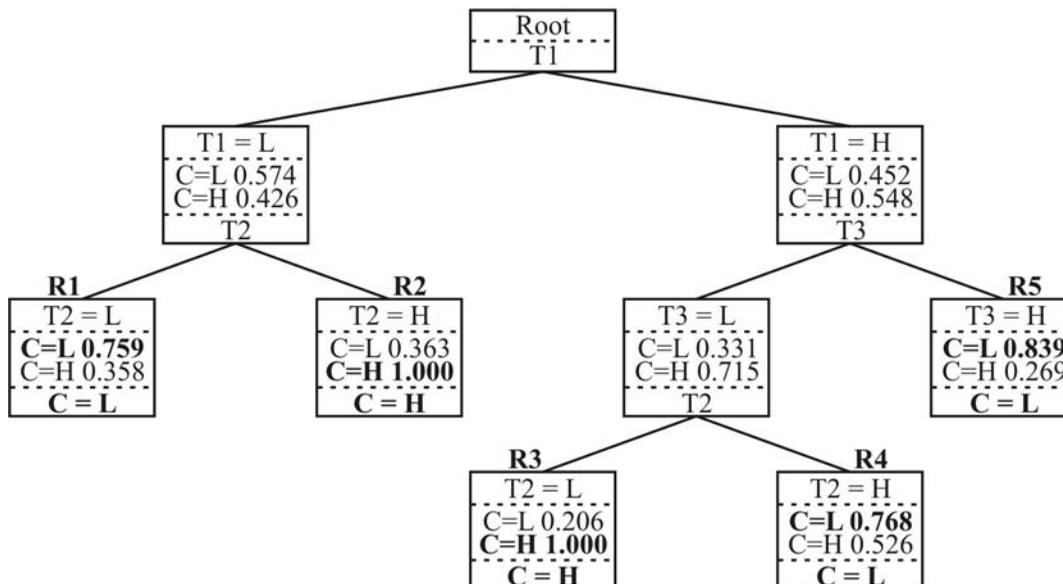
environment allows for the potential critical effects of imprecision to be mitigated, as well as brings a beneficial level of interpretability to the results found, through the decision rules defined.

As with the more traditional 'crisp' decision tree approaches, there are issues such as the complexity of the results, in this case the tree defined. Future trends will surely include how FDTs can work on re-grading the complexity of the tree constructed, commonly known as pruning. Further, the applicability of the rules constructed, should see the use of FDTs extending in the range of applications it can work with.

CONCLUSION

The interest in FDTs, with respect to AI, is due to the inductive learning and linguistic rule construction processes that are inherent with it. The induction undertaken, truly lets the analysis create intelligent results from the data available. Many of the applications FDTs have been used within, such medicine, have benefited greatly from the interpretative power of the readable rules. The accessibility of the results from FDTs should secure it a positive future.

REFERENCES

Armand, S., Watelain, E., Roux, E., Mercier, M., & Lepoutre, F.-X. (2007). Linking clinical measurements and kinematic gait patterns of toe-walking using fuzzy decision trees. *Gait & Posture*, 25(3), 475-484.

Bhatt, R.B., & Gopal, M. (2005). Improving the Learning Accuracy of Fuzzy Decision Trees by Direct Back Propagation, *IEEE International Conference on Fuzzy Systems*, 761-766.

Chang, R. L. P., & Pavlidis, T. (1977). Fuzzy decision tree algorithms. *IEEE Transactions Systems Man and Cybernetics*, *SMC-7*(1), 28-35.

Chiang, I.-J., & Hsu, J.Y.-J. (2002). Fuzzy classification trees for data analysis. *Fuzzy Sets and Systems*, *130*, 87-99.

Dombi, J., & Gera, Z. (2005). The approximation of piecewise linear membership functions and Łukasiewicz

operators. *Fuzzy Sets and Systems*, *154*, 275-286.

Hunt, E.B., Marin, J., & Stone, P.T. (1966). *Experiments in Induction*. New York, NY: Academic Press.

Li, M.-T., Zhao, F., & Chow L.-F. (2006). Assignment of Seasonal Factor Categories to Urban Coverage Count Stations Using a Fuzzy Decision Tree. *Journal of Transportation Engineering*, *132*(8), 654-662.

Pal, N.R., & Chakraborty, S. (2001). Fuzzy Rule Extraction From ID3-Type Decision Trees for Real Data. *IEEE Transactions on Systems, Man, and Cybernetics B*, *31*(5), 745-754.

Quinlan, J.R. (1979). Discovery rules by induction from large collections of examples, in: D. Michie (Ed.), Expert Systems in the Micro Electronic Age, Edinburgh University Press, Edinburgh, UK.

Quinlan, J.R. (1987). Probabilistic decision trees, in: P. Langley (Ed.), Proc. 4th Int. Workshop on Machine Learning, Los Altos, CA.

Wang, X., Nauck, D.D., Spott, M., & Kruse, R. (2007). Intelligent data analysis with fuzzy decision trees. *Soft Computing*, *11*, 439-457.

Yuan, Y., & Shaw, M.J. (1995). Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, *69*(2), 125-139.

Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control, 8*(3), 338-353.

KEY TERMS

Condition Attribute: An attribute that describes an object. Within a decision tree it is part of a non-leaf node, so performs as an antecedent in the decision rules used for the final classification of an object.

Decision Attribute: An attribute that characterises an object. Within a decision tree is part of a leaf node, so performs as a consequent, in the decision rules, from the paths down the tree to the leaf node.

Decision Tree: A tree-like structure for representing a collection of hierarchical decision rules that lead to a class or value, starting from a root node ending in a series of leaf nodes.

F

Induction: A technique that infers generalizations from the information in the data.

Leaf Node: A node not further split, the terminal grouping, in a classification or decision tree.

Linguistic Term: One of a set of linguistic terms, which are subjective categories for a linguistic variable, each described by a membership function.

Linguistic Variable: A variable made up of a number of words (linguistic terms) with associated degrees of membership.

Path: A path down the tree from root node to leaf node, also termed a branch.

Membership Function: A function that quantifies the grade of membership of a variable to a linguistic term.

Node: A junction point down a path in a decision tree that describes a condition in an if-then decision rule. From a node, the current path may separate into two or more paths.

Root Node: The node at the tope of a decision tree, from which all paths originate and lead to a leaf node.

Fuzzy Graphs and Fuzzy Hypergraphs

Leonid S. Bershtein
Taganrog Technological Institute of Southern Federal University, Russia

Alexander V. Bozhenyuk
Taganrog Technological Institute of Southern Federal University, Russia

INTRODUCTION

Graph theory has numerous application to problems in systems analysis, operations research, economics, and transportation. However, in many cases, some aspects of a graph-theoretic problem may be uncertain. For example, the vehicle travel time or vehicle capacity on a road network may not be known exactly. In such cases, it is natural to deal with the uncertainty using the methods of fuzzy sets and fuzzy logic.

Hypergraphs (Berge,1989) are the generalization of graphs in case of set of multiarity relations. It means the expansion of graph models for the modeling complex systems. In case of modelling systems with fuzzy binary and multiarity relations between objects, transition to fuzzy hypergraphs, which combine advantages both fuzzy and graph models, is more natural. It allows to realise formal optimisation and logical procedures.

However, using of the fuzzy graphs and hypergraphs as the models of various systems (social, economic systems, communication networks and others) leads to difficulties. The graph isomorphic transformations are reduced to redefinition of vertices and edges. This redefinition doesn't change properties the graph determined by an adjacent and an incidence of its vertices and edges.

Fuzzy independent set, domination fuzzy set, fuzzy chromatic set are invariants concerning the isomorphism transformations of the fuzzy graphs and fuzzy hypergraph and allow make theirs structural analysis.

BACKGROUND

The idea of fuzzy graphs has been introduced by Rosenfeld in a paper in (Zadeh, 1975), which has also been discussed in (Kaufmann, 1977).

The questions of using fuzzy graphs for cluster analysis were considered in (Matula,1970, Matula,1972). The questions of using fuzzy graphs in Database Theory were discussed in (Kiss,1991). The tasks of allocations centers on fuzzy graphs were considered in (Moreno, Moreno & Verdegay, 2001, Kutangila-Mayoya & Verdegay, 2005, Rozenberg & Starostina, 2005). The analyses and research of flows and vitality in transportation nets were considered in (Bozhenyuk, Rozenberg & Starostina, 2006). The fuzzy hypergraph applications to portfolio management, managerial decision making, neural cell-assemblies were considered in (Monderson & Nair, 2000). The using of fuzzy hypergraphs for decision making in CAD-Systems were also considered in (Malyshev, Bershtein & Bozhenyuk, 1991).

MAIN DEFINITIONS OF FUZZY GRAPHS AND HYPERGRAPHS

This article presents the main notations of fuzzy graphs and fuzzy hypergraphs, invariants of fuzzy graphs and hypergraphs.

Fuzzy Graph

Let a fuzzy direct graph $\widetilde{G} = (X, \widetilde{U})$ is given, where X is a set of **vertices,** $\widetilde{U} = \{\mu_U(x_i, x_j)|(x_i, x_j) \in X^2\}$ is a fuzzy set of **edges** with the membership function $\mu_U : X^2 \rightarrow [0,1]$ (Kaufmann, 1977).

Example 1. Let fuzzy graph $\widetilde{G}$ has X={x_1, x_2, x_3, x_4}, and $\widetilde{U}$ ={<0.5/(x_1, x_2)>, <0.6/(x_2, x_3)> , <0.3/(x_3, x_4)> , <0.2/(x_4, x_1)> , <1/(x_4, x_2)>}. It is presented in figure 1.

The fuzzy graph $\widetilde{G}$ may present a fuzzy dependence relation between objects x_1, x_2, x_3, end x_4. If the object x_j fuzzy depends from the object x_i, then there is direct **edge** (x_i, x_j) with membership function $\mu_U(x_i, x_j)$.

If a fuzzy relation, presented by fuzzy graph $\widetilde{G}$, is symmetrical, we have the fuzzy nondirect graph.

Figure 1.
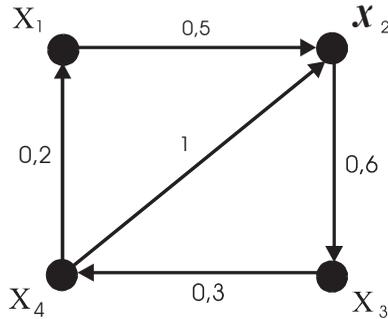
A fuzzy graph $\tilde{G} = (X,\tilde{U})$ is convenient for representing as fuzzy **adjacent matrix** $\left\|r_{ij}\right\|_{n\times n}$, where $r_{ij} = \mu_U(x_i, x_j)$. So, the fuzzy graph, presented in figure 1, may be consider by adjacent matrix:

$$R_X = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ 0 & 0,5 & 0 & 0 \\ 0 & 0 & 0,6 & 0 \\ 0 & 0 & 0 & 0,3 \\ 0,2 & 1 & 0 & 0 \end{array}.$$

The fuzzy graph $\tilde{H} = (X',\tilde{U}')$ is called a fuzzy **subgraph** (Monderson & Nair, 2000) of $\tilde{G} = (X,\tilde{U})$ if $X' \subseteq X$ and $\tilde{U}' \subseteq \tilde{U}$.

Fuzzy directed **path** (Bershtein & Bozhenyuk, 2005) $\tilde{L}(x_i,x_m)$ of graph $\tilde{G} = (X,\tilde{U})$ is called the sequence of fuzzy directed edges from vertex x_i to vertex x_m:

$$\tilde{L}(x_i, x_m) = < \mu_U(x_i,x_j)/(x_i,x_j) >, < \mu_U(x_j,x_k)/(x_j,x_k) >, ...$$
$$..., < \mu_U(x_1,x_m)/(x_1,x_m) >.$$

Conjunctive strength of path $\mu(\tilde{L}(x_i,x_m))$ is defined as:

$$\mu(\tilde{L}(x_i,x_m)) = \underset{<x_\alpha,x_\beta>?\tilde{L}(x_i,x_m)}{\&} \mu_U < x_\alpha,x_\beta >.$$

Fuzzy directed path $\tilde{L}(x_i,x_m)$ is called simple path between vertices x_i and x_m if its part is not a path between the same vertices.

If a number of vertices $n \geq 3$ and $x_i = x_m$, then the path is called a cycle.

Obviously, what is it definition coincides with the same definition for nonfuzzy graphs.

Vertex y is called fuzzy accessible from vertex x in the graph $\tilde{G} = (X,\tilde{U})$ if exists a fuzzy directed path from vertex x to vertex y.

The accessible degree of vertex y from vertex x, $(x \neq y)$ is defined by expression:

$$\gamma(x,y) = \max_{\alpha} (\mu(\tilde{L}_\alpha(x,y)), \quad \alpha = 1,2,...,p,$$

where p - number of various simple directed paths from vertex x to vertex y.

A subset of vertices X' is called a fuzzy independent vertex set (Bershtein & Bozhenuk, 2001) with the degree of independence

$$\alpha(X') = 1 - \max_{? x_i,x_j ? X} \{\mu_U(x_i, x_j)\}.$$

A subset of vertices $X' \subseteq X$ of graph $\tilde{G}$ is called a maximal fuzzy independent **vertex** set with the degree $\alpha(X')$, if the condition $\alpha(X'') < \alpha(X')$ is true for any $X' \subset X''$.

Let a set $\tau_k = \{X_{k1}, X_{k2},...,X_{kl}\}$ be given where X_{ki} is a fuzzy independent k-vertex set with the degree of independent α_{ki}. We define as

$$\alpha_k^{max} = \textbf{max}\{\alpha_{x_k^1}, \alpha_{x_k^2}, ..., \alpha_{x_k^l}\}.$$

The value α_k^{max} means that fuzzy graph $\tilde{G}$ includes k-vertex subgraph with the degree of independent α_k^{max} and doesn't include k-vertex subgraph with the degree of independence more than α_k^{max}.

A fuzzy set

$$\Psi_X = \{< \alpha_1^{max}/1 >, < \alpha_2^{max}/2 >, ..., < \alpha_n^{max}/n >\}$$

is called a fuzzy **independent set** of fuzzy graph $\tilde{G}$.

Fuzzy graph $\tilde{G}$, presented in figure 1, has seven maximum fuzzy independent vertex sets:

$$\Psi_1 = \{x_2\}, \ \Psi_2 = \{x_4\}, \ \Psi_3 = \{x_1, x_3\}$$

with the degree of independence 1; $\Psi_4 = \{x_1, x_4\}$ with the degree of independence 0,8; $\Psi_5 = \{x_1, x_3, x_4\}$ with

the degree of independence 0,7; $\Psi_6 = \{x_1, x_2\}$ with the degree of independence 0,5 and $\Psi_7 = \{x_1, x_2, x_3\}$ with the degree of independence 0,4. So, its fuzzy independent set is defined as

$$\Psi_X = \{<1/1>,<1/2>,<0,7/3>,<0/4>\}.$$

Let X' be an arbitrary subset of the vertex set X. For each vertex $y \in X \setminus X'$ we define the value:

$$\gamma(y) = \max_{x \in X}\{\mu_U(y,x)\}.$$

The set X' is called a fuzzy dominating vertex set for vertex y with the degree of domination $\gamma(y)$.

The set X' is called a fuzzy dominating vertex set for the graph $\tilde{G}$ with the degree of domination

$$\beta(X') = \min_{y \in X \setminus X'} \max_{x \in X'}\{\mu_U(y,x)\}.$$

A subset $X' \subseteq X$ of graph $\tilde{G}$ is called a minimal fuzzy dominating **vertex** set with the degree $\beta(X')$ if the condition $\beta(X'') < \beta(X')$ is true for any subset $X'' \subset X'$.

Let a set $\tau_k = \{X_{k1}, X_{k2},...,X_{kl}\}$ be given, where X_{ki} is a fuzzy dominating k-vertex set with the degree of domination β_{ki}. We define as $\beta_k^{min} = \max\{\beta_{X_k^1}, \beta_{X_k^2},..., \beta_{X_k^l}\}$. In the case $\tau_k = \varnothing$ we define $\beta_{X_k}^{min} = \beta_{X_{k-1}}^{min}$. Volume β_k^{min} means that fuzzy graph $\tilde{G}$ includes k-vertex **subgraph** with the degree of domination β_k^{min} and doesn't include k-vertex subgraph with the degree of domination more than β_k^{min}.

A fuzzy set $\tilde{B}_X = \{<b_1^{min}/1>,<b_2^{min}/2>,...,<b_n^{min}/n>\}$ is called a **domination fuzzy set** of fuzzy graph $\tilde{G}$ (Bershtein & Bozhenuk, 2001 a).

Fuzzy graph $\tilde{G}$ (Figure 1) has five fuzzy minimal dominating vertex sets: $P_1 = \{x_1, x_2, x_3\}$ with the degree of domination 1; $P_2 = \{x_1, x_3, x_4\}$ with degree of domination 0,6; $P_3 = \{x_2, x_3\}$ with the degree of domination 0,5; $P_4 = \{x_1, x_3\}$ with degree of domination 0,2 and $P_5 = \{x_2, x_4\}$ with the degree of domination 0,3. A domination fuzzy set of fuzzy graph $\tilde{G}$ is defined as $\tilde{B}_X = \{<0/1>,<0,5/2>,<1/3>,<1/4>\}$.

A value

$$L = \underset{i\equiv 1,k}{\&} \alpha_i = \underset{i\equiv 1,k}{\&} (1 - \underset{x,y \in X_i}{\vee} \mu_G(x,y))$$

is called a separation degree of fuzzy graph $\tilde{G}$ with k colors.

The fuzzy graph $\tilde{G}$ may be colored in a number of colors from 1 to n. In this case the separation degree L depends of the number of colors. For the fuzzy graph $\tilde{G}$ we relate a family of fuzzy sets

$$\mathfrak{R} = \{\tilde{A}_G\}, \ \tilde{A}_G = \{<L_{\tilde{A}}(k)/k|k = \overline{1,n}\}$$

where $L_{\tilde{A}}(k)$ defines a degree of separation of fuzzy graph $\tilde{G}$ with k colors.

A fuzzy set $\tilde{\gamma} = \{<L_{\tilde{\gamma}}(k)/k|k = \overline{1,n}\}$ is called a **fuzzy chromatic set** of graph $\tilde{G}$ if the condition $\tilde{A}_G \subseteq \tilde{\gamma}$ is performed for any set $\tilde{A}_G \in \mathfrak{R}$, or else: $(\forall \tilde{A}_G \in \mathfrak{R})(\forall k = \overline{1,n})[L_A(k) \le L_{\tilde{\gamma}}(k)]$ (Bershtein & Bozhenuk, 2001 b).

Otherwise, the fuzzy chromatic set defines a maximal separation degree of fuzzy graph $\tilde{G}$ with k= 1, 2,..., n colors.

For fuzzy graph $\tilde{G}$ (Figure 1) the fuzzy chromatic set is $\tilde{\gamma}(\tilde{G}) = \{<0/1>,<0,5/2>,<1/3>\}$.

So, the fuzzy graph $\tilde{G}$ may be colored

- by one color with the degree of separation 0. In other words, there is at least pair of vertices x_i and x_j for which the membership function $\mu_U(x_i, x_j) = 1$. In our graph, these vertices are x_4 and x_2;
- by 2 colors with the degree of separation 0,5 (vertices x_1, x_2 - first color, vertices x_3, x_4 - second color). In other words, between vertices of the same color there aren't edges with the membership function more than 0,5;
- by 3 colors with the degree of separation 1 (vertices x_1, x_3, - first color, vertices x_2 - second color, vertex x_4 - third color). In other words, between vertices of the same color there aren't any edges.

Fuzzy Hypergraph

Let a fuzzy hypergraph $\tilde{H} = (X, \tilde{E})$ be given, where $X = \{x_i\}$, $i \in I = \{1,2,...,n\}$ – is a finite set and $\tilde{E} = \{\tilde{e}_k\}$, $\tilde{e}_k = \{<\mu_{e_k}(x)/x>\}$, $k \in K = \{1,2,...,m\}$ is a family of fuzzy subsets in X (Monderson & Nair, 2000, Bershtein & Bozhenyuk, 2005). Thus elements of set X are the vertices of hypergraph, a family $\tilde{E}$ is the family of hypergraph fuzzy edges. The value $\mu_{e_k}(x) \in [0,1]$ is an incidence degree of a vertex x to an edge $\tilde{e}_k$.

It is possible to see that a fuzzy hypergraph is turned in the fuzzy graph when $1 \leq |\tilde{e}_k| \leq 2, k \in K$.

Vertices x and y are called fuzzy adjacent vertices if there are some edge, which includes both vertices. In this case a value

$$\mu(x,y) = \vee_{e_k ? \tilde{E}} \mu_{e_k}(x) \& \mu_{e_k}(y)$$

is called an adjacent degree of two vertices x and y of fuzzy hypergraph $\tilde{H}$.

Two edges $\tilde{e}_i$ and $\tilde{e}_j$ are called fuzzy adjacent edges if $\tilde{e}_i \cap \tilde{e}_j \neq \varnothing$. In this case a value

$$\mu(\tilde{e}_i, \tilde{e}_j) = \bigvee_{x \in (e_i \cap e_j)} \mu_{e_i \cap e_j}(x)$$

is called adjacent degree of edges $\tilde{e}_i$ and $\tilde{e}_j$.

A fuzzy hypergraph $\tilde{H} = (X, \tilde{E})$ is convenient for representing as **fuzzy incidence matrix** $\|r_{\tilde{j}}\|_{n \times m}$, where $r_{\tilde{j}} = m_{e_j}(x_i)$. So, any matrix, which elements are included in the interval [0,1], may be consider as fuzzy incidence matrix of some fuzzy hypergraph.

A fuzzy simple **path** $\tilde{C}(x_1, x_{q+1})$ with the length q is defined as the sequence

$$\tilde{C}(x_1, x_{q+1}) = (x_1, \mu_{e_1}(x_1), \tilde{e}_1, \mu_{e_1}(x_2), x_2, \mu_{e_2}(x_2),$$
$$\tilde{e}_2, \ldots, \tilde{e}_q, \mu_{e_q}(x_{q+1}), x_{q+1}),$$

where all vertices $x_1, \ldots, x_q \in X$ and all edges $\tilde{e}_1, \ldots, \tilde{e}_q \in \tilde{E}$ are different.

A strength of fuzzy simple path is the weakest of adjacent degrees, which are included in this path $\tilde{C}(x_1, x_{q+1})$. If two vertices x_1 and x_{q+1} are connected by paths $\tilde{C}_1, \tilde{C}_2, \ldots \tilde{C}_t$ with strengths $\mu_1, \mu_2, \ldots \mu_t$, then say that vertices x_1 and x_{q+1} are fuzzy connected by the strength $\mu(x_1, x_{q+1}) = \mu_1 \vee \mu_2 \vee \ldots \vee \mu_t$.

An internal stability degree of vertices subset X' of fuzzy hypergraph $\tilde{H}$ is determined as:

$$\alpha_{X'} = 1 - \max_{x,y \in X'} \mu(x,y).$$

Subset $X' \subseteq X$ is called a maximal fuzzy internally stable set with the degree of internal stability a_X, if the statement $(\forall X'' \supseteq X')(\alpha_{X''} < \alpha_{X'})$ is true.

Let's paint each vertex $x \in X$ of hypergraph $\tilde{H}$ in one of k colours ($1 \leq k \leq n$) and we shall consider a X_i, subset of vertices, colored identically.

The value

$$L_I = \&_{i \equiv 1, k} \alpha_i = \&_{i \equiv 1, k} (1 - \vee_{x,y \in X_i} \mu(x,y))$$

is called a separation degree of fuzzy hypergraph $\tilde{H}$ at its k-colorings (Bershtein, Bozhenuk & Rozenberg, 2005)..

Fuzzy hypergraph $\tilde{H}$ can be colored in any number of k colours and thus separation degree L depends on their number. Fuzzy hypergraph $\tilde{H}$ we shall put in conformity family of fuzzy sets

$$\mathfrak{R} = \{\tilde{A}_{\tilde{H}}\}, \tilde{A}_{\tilde{H}} = \{< L(k) / k \mid k = \overline{1,n}\},$$

where $L(k)$ determines a separation degree of fuzzy hypergraph $\tilde{H}$ at its certain k - colouring.

Fuzzy set $\tilde{\gamma} = \{< L_{\tilde{\gamma}}(k) / k \mid k = \overline{1,n}\}$ is called **fuzzy chromatic set** of hypergraph $\tilde{H}$, if for any other set $\tilde{A}_{\tilde{H}} \in \mathfrak{R}$, it is true $\tilde{A}_{\tilde{H}} \subseteq \tilde{\gamma}$. In other words, $(\forall \tilde{A}_{\tilde{H}} \in \mathfrak{R})(\forall k = \overline{1,n}) L(k) \leq L(k)$. Or, otherwise, fuzzy chromatic set of hypergraph $\tilde{H}$ determines the greatest separation degrees at colouring its tops in one of 1,2...n colours.

Let $\tilde{H}$ be a fuzzy hypergraph which the incidence matrix is given by:

$$I = \begin{array}{c|ccccc} & \tilde{e}_1 & \tilde{e}_2 & \tilde{e}_3 & \tilde{e}_4 & \tilde{e}_5 \\ \hline x_1 & 0,8 & 0,5 & 0 & 0 & 0 \\ x_2 & 1 & 0 & 0 & 0 & 0 \\ x_3 & 0,4 & 1 & 0,3 & 0,7 & 0 \\ x_4 & 0 & 0,6 & 0,4 & 0,2 & 1 \\ x_5 & 0 & 0 & 0,7 & 1 & 0 \\ x_6 & 0 & 0 & 0 & 0 & 0,4 \end{array}$$

The fuzzy chromatic set for the fuzzy hypergraph is

$$\tilde{\gamma}_I = \{< 0,2 / 1 >, < 0,5 / 2 >, < 1 / 3 >\}.$$

Otherwise, the fuzzy hypergraph may be colored by one color with the degree of separation 0,2; by 2 colors with the degree of separation 0,5 (vertices x_2, x_3 and x_6 - first color, vertices x_1, x_4 и x_5 - second color); by 3 colors with the degree of separation 1 (vertices x_2, and x_4 - first color, vertices x_1, x_5 and x_6 - second color, vertex x_3 - third color).

FUTURE TRENDS

In according to a principle of generalization L. Zadeh, the theory of fuzzy graphs and fuzzy hypergraphs will develop in a development course of nonfuzzy graphs, hypergraphs, and fuzzy sets theory.

CONCLUSION

When we consider fuzzy graphs and fuzzy hypergraphs, there is an opportunity to relate any set vertices and edges to family of partial graphs and hypergraphs with given property. For example, a sequence of edges – to family of graph paths; a sequence of vertices and edges – to family of bipartite graphs, and so on. It enables to define new properties of fuzzy graphs and hypergraphs, and to use theirs to analysis and synthesis fuzzy systems.

REFERENCES

Berge, C. (1989). *Hypergraphs: combinatorics of finite sets*. Elsevier Science Publishers.

Bershtein, L.S. & Bozhenuk A.V. (2001 a). Maghout Method for Determination of Fuzzy Independent, Dominating Vertex Sets and Fuzzy Graph Kernels. *J. General Systems,* 30, 45-52.

Bershtein, L.S. & Bozhenyuk A.V. (2001 b). A Color Problem for Fuzzy Graph. *Computation intelligence: theory and applications; international conference; proceedings. 7th Fuzzy Days, Dortmund, Germany, October 1-3, 2001. Bernd Reusch (ed.): Springer-Verlag* (2206). 500-505.

Bershtein, L.S. & Bozhenyuk, A.V. (2005). *Fuzzy Graphs and Hypergraphs.* Moscow, Nauchniy Mir.

Bershtein, L.S., Bozhenyuk, A.V. & Rozenberg, I.N. (2005). Fuzzy Coloring of Fuzzy Hypergraph. *Computation Intelligence, Theory and Applications. International Conference 8th Fuzzy Days in Dortmund, Germany, Sept. 29- Oct. 01, 2004 Proceedings. Bernd Reusch (ed.): Springer-Verlag.* 703-711.

Bozhenyuk, A.V., Rozenberg, I.N. & Starostina, T.A. (2006). *Analysis and Research of Flows and Vitality in Transportation Nets with Fuzzy Dates*. Moscow, Nauchniy Mir.

Kaufmann, A. (1977). *Introduction a la theorie des sous-ensemles flous*, Masson, Paris, France.

Kiss, A. (1991). An Application of Fuzzy Graphs in Database Theory, Automata, Languages and Programming Systems. Pure Math., Appl. Ser. A, 1, 337-342.

Kutangila-Mayoya, D. & Verdegay, J.L. (2005). P-Median Problems in a Fuzzy Environment. *Mathware & Soft Computing*, 12, 97-106.

Malyshev, N.G., Bershtein, L.S. & Bozhenyuk, A.V. (1991). *Fuzzy Models for Expert Systems in CAD-Systems*. Moscow, Energoatomizdat.

Matula, D.W. (1970). Cluster Analysis Via Graph Theoretic Techniques: *Proc. of Lousiana Conf. on Combinatorics, Graph Theory and Computing*. 199-212.

Matula, D.W. (1972). K-components, Clusters, and Slicings in Graphs. *SIAM J. Appl. Math*., 22, 459-480.

Monderson, J.N. & Nair, P.S. (2000). *Fuzzy Graphs and Fuzzy Hypergraphs*. Heidelberg; New-York: Physica-Verl.

Moreno Perez, J.A., Moreno-Vega, J.M. & Verdegay, J.L. (2001). In Location Problem on Fuzzy Graphs. *Mathware & Soft Computing*, 8, 217-225.

Zadeh, L.A. (1975). Fuzzy sets and their application to cognitive and decision, Academic Press, New York, USA.

KEY TERMS

Binary Relation: A binary relation R from a set A to a set B is a subset of $A \times B$.

Binary Symmetric Relation: A relation R on a set A is symmetric if for all $x, y \in A$ $xRy \Rightarrow yRx$.

Fuzzy Set: A generalization of the definition of the classical set. A fuzzy set is characterized by a membership function, which maps the member of the universe into the unit interval, thus assigning to elements of the universe degrees of belongingness with respect to a set.

Graph: A graph $G = (V, E)$ is a mathematical structure consisting of two finite sets V and E. The elements of V are called vertices (or nodes), and the elements of E are called edges. Each edge has a set of one or two vertices associated to it, which are called its endpoints.

Graph Invariant: A property of a graph that is preserved by isomorphisms.

Isomorphic Graphs: Two graphs that have a structure-preserving vertex bijection between them.

Hypergraph: A hypergraph on a finite set $X=\{x_1,x_2,...,x_n\}$ is a family $H=\{E_1,E_2,...,E_m\}$ of subsets of X such that $E_i \neq \varnothing$ and

$$\bigcup_{i=1}^{m} E_i = X$$
.

Membership Function: The membership function of a fuzzy set is a generalization of the characteristic function of crisp sets.

Multiarity Relation: A multiarity relation R between elements of sets A, B, …, C is a subset of $A \times B \times ... \times C$.

F

Fuzzy Logic Applied to Biomedical Image Analysis

Alfonso Castro
University of A Coruña, Spain

Bernardino Arcay
University of A Coruña, Spain

INTRODUCTION

Ever since Zadeh established the basis of fuzzy logic in his famous article Fuzzy Sets (Zadeh, 1965), an increasing number of research areas have used his technique to solve and model problems and apply it, mainly, to control systems. This proliferation is largely due to the good results in classifying the ambiguous information that is typical of complex systems. Success in this field has been so overwhelming that it can be found in many industrial developments of the last decade: control of the Sendai train (Yasunobu & Miyamoto, 1985), control of air-conditioning systems, washing machines, auto-focus in cameras, industrial robots, etc. (Shaw, 1998)

Fuzzy logic has also been applied to computerized image analysis (Bezdek & Keller & Krishnapuram & Pal, 1999) because of its particular virtues: high noise insensitivity and the ability to easily handle multidimensional information (Sutton & Bezdek & Cahoon, 1999), features that are present in most digital images analyses. In fuzzy logic, the techniques that have been most often applied to image analysis have been fuzzy clustering algorithms, ever since Bezdek proposed them in the seventies (Bezdek, 1973). This technique has evolved continuously towards correcting the problems of the initial algorithms and obtaining a better classification: techniques for a better initialization of these algorithms, and algorithms that would allow the evaluation of the solution by means of validity functions. Also, the classification mechanism was improved by modifying the membership function of the algorithm, allowing it to present an adaptative behaviour; recently, kernel functions were applied to the calculation of memberships. (Zhong & Wei & Jian, 2003)

At the present moment, applications of fuzzy logic are found in nearly all Computer Sciences fields, it constitutes one of the most promising branches of Artificial Intelligence both from a theoretic and commercial point of view. A proof of this evolution is the development of intelligent systems based on fuzzy logic.

This article presents several fuzzy clustering algorithms applied to medical images analysis. We also include the results of a study that uses biomedical images to illustrate the mentioned concepts and techniques.

BACKGROUND

Fuzzy logic is an extension of the traditional binary logic that allows us to achieve multi-evaluated logic by describing domains in a much more detailed manner and by classifying better through searches in a more extensive area. Fuzzy logic makes it possible to model the real world more efficiently: for example, whereas binary logic merely allows us to state that a coffee is hot or cold, fuzzy logic allows us to distinguish between all the possible temperature fluctuations: very hot, lukewarm, cold, very cold, etc.

Techniques based on fuzzy logic have proven to be very useful for dealing with the ambiguity and vagueness that are normally associated to digital images analysis. At what grey level do we fixate the thresholding? Where do we locate the edge in blurred objects? When is a grey level high, low, or average?

The fuzzy processing of digital images can be considered a totally different focus with respect to the traditional computerized vision techniques. It was not developed to solve a specific problem, but describes a new class of image processing techniques and a new methodology to develop them: fuzzy edge detectors, fuzzy geometric operators, fuzzy morphological operators, etc.

These features make fuzzy logic especially useful for the development of algorithms that improve medical images analysis, because it provides a framework

for the representation of knowledge that can be used in any phase of the analysis. (Wu & Agam & Roy & Armato, 2004) (Vermandel & Betrouni & Taschner & Vasseu & Rosseau, 2007)

FUZZY CLUSTERING ALGORITHMS APPLIED TO BIOMEDICAL IMAGE ANALYSIS

Medical imaging systems use a series of sensors that detect the features of the tissues and the structure of the organs, which allows us, depending on the used technique, to obtain a great amount of information and images of the area from different angles. These virtues have converted them into one of the most popular support techniques in diagnosis, and have given rise to the current distribution and variety in medical images modalities (X-Rays, PET …) and to new modalities that are being developed (fMRI).

The complexity of the segmentation of biomedical images is entirely due to its characteristics: the large amount of data that need to be analyzed, the loss of information associated to the transition from a 3D body to a 2D representation, the great variability and complexity of the shapes that must be analyzed … Among the most frequently applied focuses to segment medical images is the use of pattern recognition techniques, since normally the purpose of analyzing a medical digital image is the detection of a particular element or object: tumors, organs, etc.

Of all these techniques, fuzzy clustering techniques have proven to be among the most powerful ones, because they allow us to use several features of the dataset, each with their own dimensionality, and to partition these data; also, they work automatically and usually have low computational requirements. Therefore, if the problem of segmentation is defined as the partition of the image into regions that have a common feature, fuzzy clustering algorithms carry out this partition with a set of exemplary elements, called centroids, and obtain a matrix of the size of the original image and with a dimensionality equal to the number of clusters into which the image was divided; this indicates the membership of each pixel to each cluster and serves as a basis for the detection of each element.

In the next section we present a series of fuzzy clustering algorithms that can be considered to reflect the evolution in this field and its various viewpoints.

Finally, these algorithms will be used in a study that shows the use and possibilities of fuzzy logic in the analysis of biomedical images.

Fuzzy C-Means (FCM)

The FCM algorithm was developed by Bezdek (Bezdek, 1973) and is the first fuzzy clustering algorithm; it initially needs the number of clusters in which the image will be divided and a sample of each cluster. The steps of this algorithm are the following:

1. Calculation of the membership of each element to each cluster:

$$u_k(i,j) = \left(\sum \frac{\left\| y(i,j) - v_k \right\|^{\frac{2}{m-1}}}{\left\| y(i,j) - v_j \right\|} \right)^{-1}$$

(1)

2. Calculation of the new centroids of the image:

$$v_k = \frac{\sum_{i,j} u_k(i,j)^m \, y(i,j)}{\sum_{i\,j} u_k(i,j)^m}, k = 1,\ldots,C$$

(2)

3. If the error stays below a determined threshold, stop. In the contrary case, return to step 1.

The parameters that were varied in the analysis of the algorithm were the provided samples and the value of *m*.

Fuzzy K-Nearest Neighbour (FKNN)

The Fuzzy K-Nearest Neighbour (Givens Jr. & Gray & Keller, 1992) is, as its name indicates, a fuzzy variant of a hard segmentation algorithm. It needs to know the number of classes into which the set that must be classified will be divided.

The element that must be classified is associated to the class of the nearest sample among the *K* most similar ones. These *K* most similar samples are known as "neighbours"; if, for instance, the neighbours are classified from more to less similar, the destination class of the studied element will be the class of the neighbour that is first on the list.

We use the expression in Equation 3 to calculate the membership factors of the pixel to the considered clusters:

$$u_i(x) = \frac{\sum_{j=1}^{K} u_{ij} \left(\frac{1}{\|x - x_j\|^{\frac{2}{m-1}}} \right)}{\sum_{j=1}^{K} \left(\frac{1}{\|x - x_j\|^{\frac{2}{m-1}}} \right)} \tag{3}$$

where u_{ij} represents the membership factor of the *j*-th sample to the *i*-th class; x_j represents one of the *K* samples that are most similar to the treated pixel; *x* represents the pixel itself; *m* is a weight factor of the distance between the pixel and the samples and $u_i(x)$ represents the level of membership of the pixel *x* to class *i*.

During the analysis of this algorithm, the parameters that varied were the samples provided as initial centroids and the considered number of neighbours.

Modified Fuzzy C-Means

This algorithm is based on the work of Young Won Lim and Sang Uk Lee (Lee & Lim, 1990), who describe an algorithm for the segmentation of color images through the study of the histograms of each color band. This algorithm also relies on the classification algorithm fuzzy c-means.

The MFCM consists of two parts:

1. A hard part that studies the histograms of an image in order to obtain the number of classes, and carries out a first global classification of the image; and
2. A fuzzy part that classifies the pixels that have more difficulties in determining the class to which they belong. The pixels of this area are called "fuzzy zone".

Once obtained the initial clusters with its centroids, the algorithm uses the FCM membership function (Eq. 2) to classify the pixels. The fuzzy points are pixels between the initial clusters and pixels of clusters too little for its consideration.

Since we do not dispose of labeled simples of each class, we use the gravity centers of the clusters to calculate the membership factors of a pixel.

During the analysis of this algorithm, we varied the value of the sigma used to smoothen the histogram, the area that the initial clusters need to survive, and the security areas around the clusters.

Kernelized Fuzzy C-Means (KFCM)

This algorithm was proposed by Wu Zhong-Dong et al (Zhong & Wei & Jian, 2003) and is based on FCM, integrated with a kernel function that allows the transfer of the data to a space with more dimensionality, which makes it easier to separate the clusters.

The most often used kernel functions are the polinomial functions (Eq. 4) and the radial base functions (Eq. 5).

$$K(X,Y) = \phi(X) \cdot \phi(Y) = (X \cdot Y + b)^d \tag{4}$$

$$K(X,Y) = \phi(X) \cdot \phi(Y) = \exp\left(-(X-Y)^2 / 2\sigma^2\right) \tag{5}$$

The algorithm consists of the following steps:

1. Calculation of the membership function:

$$u_{jk} = \frac{\left(1/d^2(X_j, V_k)\right)^{1/(q-1)}}{\sum_{j=1}^{C} \left(1/d^2(X_j, V_k)\right)^{1/(q-1)}} \tag{6}$$

where

$$d^2(X_j, V_k) = K(X_j, X_j) - 2K(X_j, V_k) + K(V_k, V_k)$$

2. Calculation of the new kernel matrix $K(X_j, \hat{V}_k)$ and $K(\hat{V}_k, \hat{V}_k)$:

$$K(X_j, \hat{V}_k) = \phi(X_j) \cdot \phi(\hat{V}_k) = \frac{\sum_{i=1}^{N} (u_{ik})^q K(X_i, X_j)}{\sum_{i=1}^{N} (u_{ik})^q} \tag{7}$$

where

$$\phi(\hat{V}_k) = \frac{\sum_{j=1}^{N} (u_{jk}) \phi(X_j)}{\sum_{j=1}^{N} (u_{jk})}$$

3. Update the memberships u_{jk} to $\hat{u}_{jk}$ by means of Equation 6.
4. If the error stays below a determined threshold, stop. In the contrary case, return to step 1.

The different parameters for the analysis of this algorithm were the initial samples.

Images Used in the Study

For the selection of the images that were used in the study (Gonzalez & Woods, 1996), we applied the traditional image processing techniques and used the histogram as basic tool. See Figure 1.

We observed that the pictures presented a high level of variation, because it was not possible to standardize the different elements that have a determining effect on them: position of the patient, luminosity, etc. We selected the pictures on the basis of a characteristic trait (bad lighting, presence of strange objects, etc.) or on their "normality" (correct lighting, good contrast, etc.). The

images were digitalized to a size of 500x500 pixels and 24 color bits per pixel, using an average scanner.

The histograms of Figure 1 show some of the characteristics that were present in most photographies. The bands with a larger amount of pixels are those of the colors red and green, because of the color of the skin and the fact that green is normally used in sanitary tissue.

The histogram is continuous and presents values in most levels, which leads us to suppose that the value of most points is determined by the combination of the three bands instead of only one band, as was to be expected. This complicates the analysis of the image with algorithms.

F

Figure 1. Photograph that was used in the study, and histogram for each color band
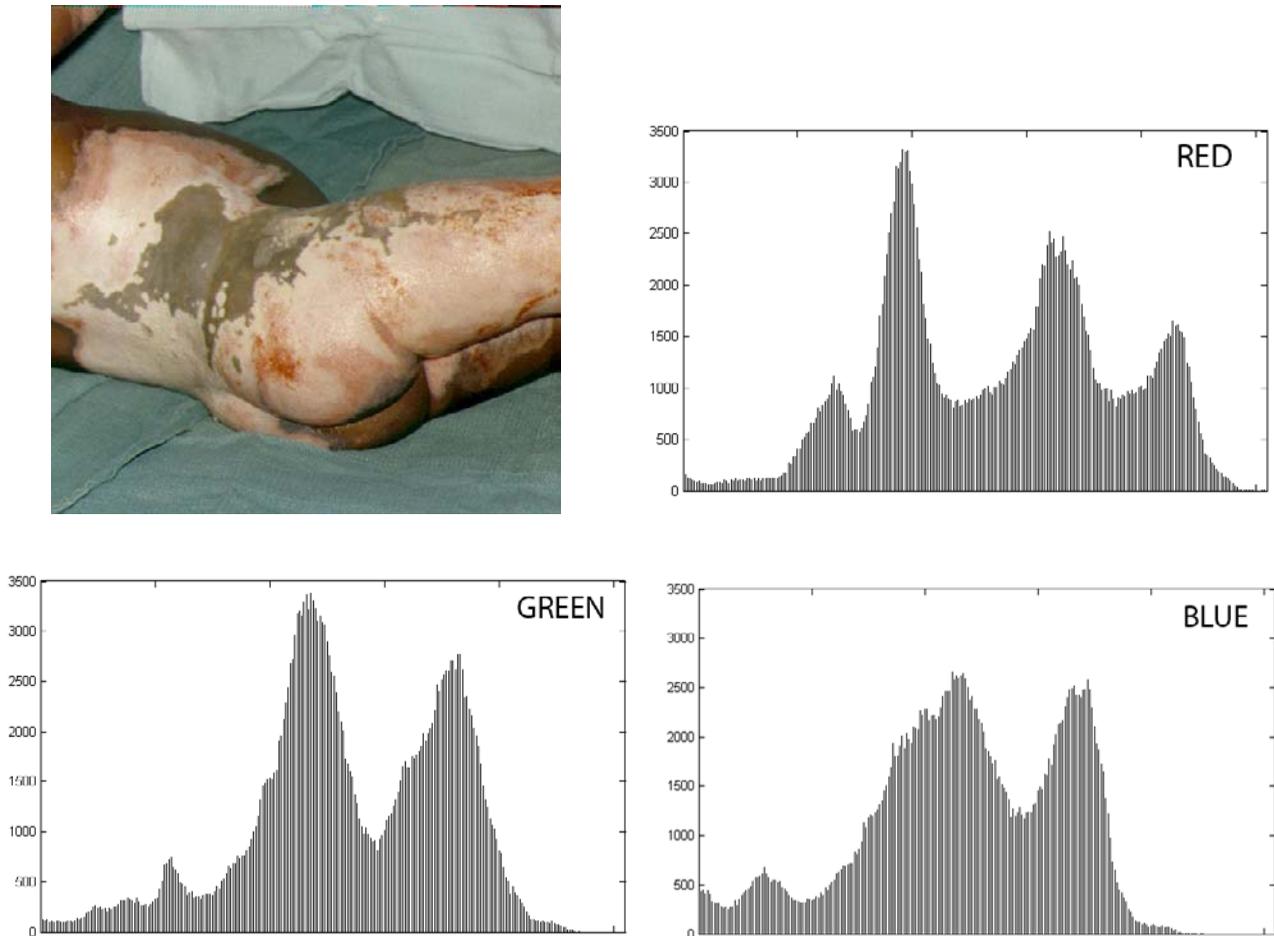
Results

The test images were divided into 3 clusters: background, healthy tissue, and burned tissue. These areas are clearly distinguished by the specialist, which allows us to build better masks to evaluate the success rate in pixel detection applied to burn wounds.

The success rate of the fuzzy clustering algorithms was first measured with Zhang's RUMA (Relative Ultimate Measurement Accuracy) (Zhang, 1996). The purpose of RUMA is to measure the quality of the segmentation in terms of the similarity of the measures

carried out on the segmented image and on the real image (Eq. 8).

$$RUMA = \frac{\left|R_f - S_f\right|}{R_f} x100\%$$

(8)

In our study, we measured the success rate by comparing the number of pixels of the burned area in the result image that coincided with pixels of the burned area in the mask.

Figure 2. Best results for the RUMA and global measurements for the: FKNN algorithm (a) and MFCM algorithm (b)

(a)

(b)
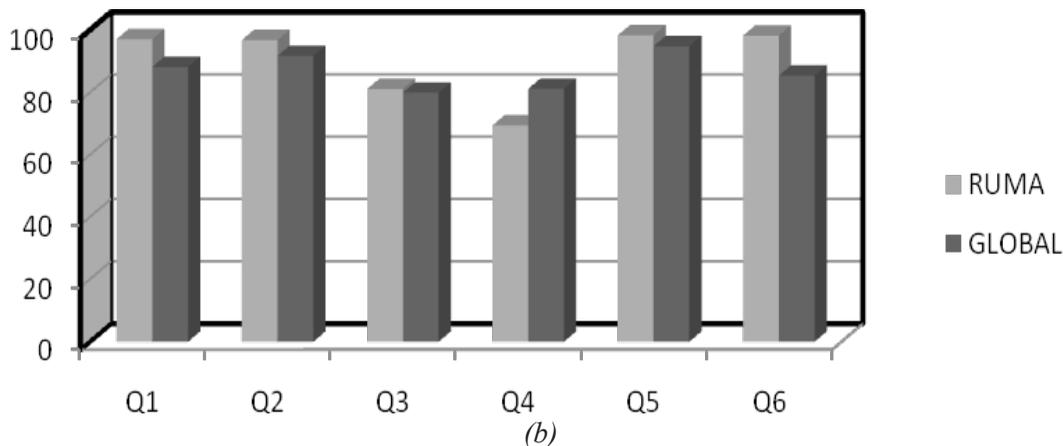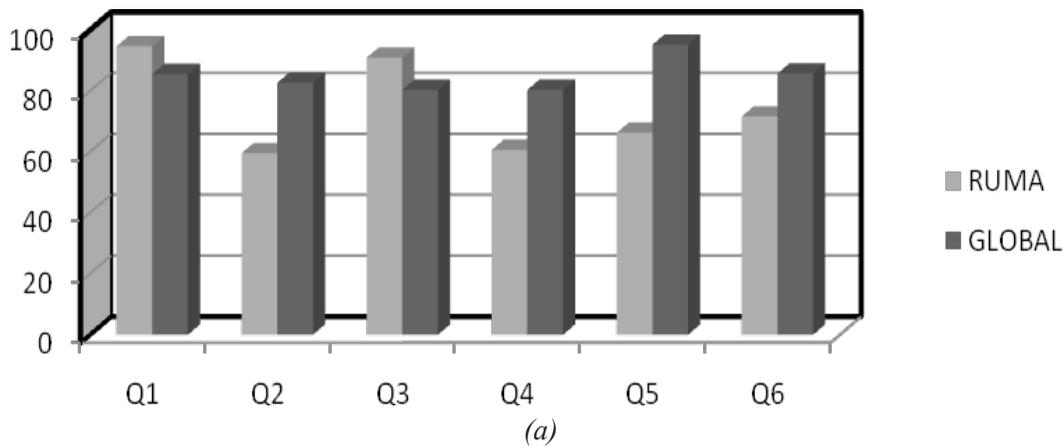
We also opted for applying a second success rate measurement, because although RUMA provides a value for the area of interest, it may not detect certain classification errors that can affect the resulting image. We use a measure that was developed by our research team and measures the clustering algorithm's performance in classifying all the pixels of the image (Eq. 9). During the development of the measure, we supposed that the error would be smaller if the error of each cluster classification were smaller, so we measured the error in the pixel classification of each cluster and weighed it against the number of pixels of that cluster.

$$error = \sum_{j=1}^{n} \sum_{i=1}^{n} \frac{F_{ij}}{MASC_j}, i \neq j \tag{9}$$

F_{ij} is the number of clusters that belong to cluster j and were assigned to cluster i, $MASC_j$ is the total amount of pixels that belong to class j, and n is the amount of clusters into which the image was divided. The value of this measurement lies between 0 and n; in order to simplify its interpretation, it was normalized between 0 and 1.

The graphics are simplified by inverting the discrepancy values: the higher the value, the better the result.

Figure 2(a) shows the best results for the FKNN algorithm, varying the number of samples and neighbours from 1 sample per cluster to 8 samples, for both measurements.

Figure 2(b) shows the results for the MFCM algorithm, varying the threshold that was required for each area in the histogram and the sigma, for both measurements.

The FCM and FKCM algorithms are not detailed because the parameters that were varied were the value of the provided samples and the stop threshold, with a rather stable result for both measurements. In the Figure 3 we can see one of the results obtained for the algorithm FCM and the imaged labeled Q1.

Figure 4(a) shows the results for the various images of the test set for RUMA applied to all the algorithms, Figure 4(b) shows the results using global measurement.

The tests reveal great variation in the values provided for the different algorithms by each measurement; this is due to the lack of homogeneous conditions in the acquisition of the images and the ensuing differences in photographic quality.

We can also observe that the results obtained with FKCM are considerably better than the results with FCM, because the first uses a better function to calculate the pixel membership. Nevertheless, for most

F

Figure 3. Image labeled Q1 (left) and one of the results obtained for the FCM algorithm (right)

Figure 4. Best results for the burned area using: RUMA measurement (a) and global success rate measurement (b)

(a)

(b)
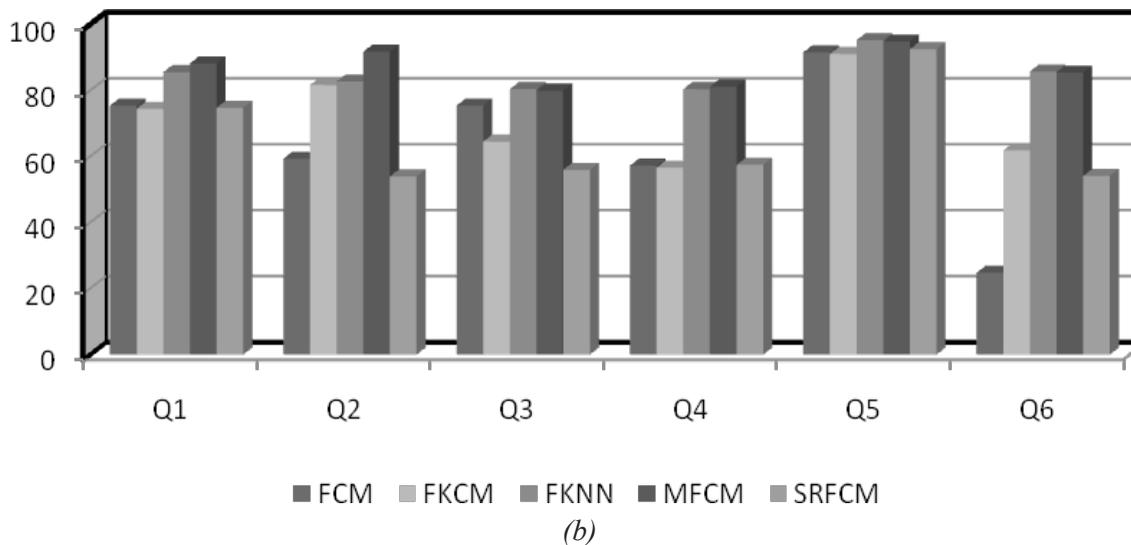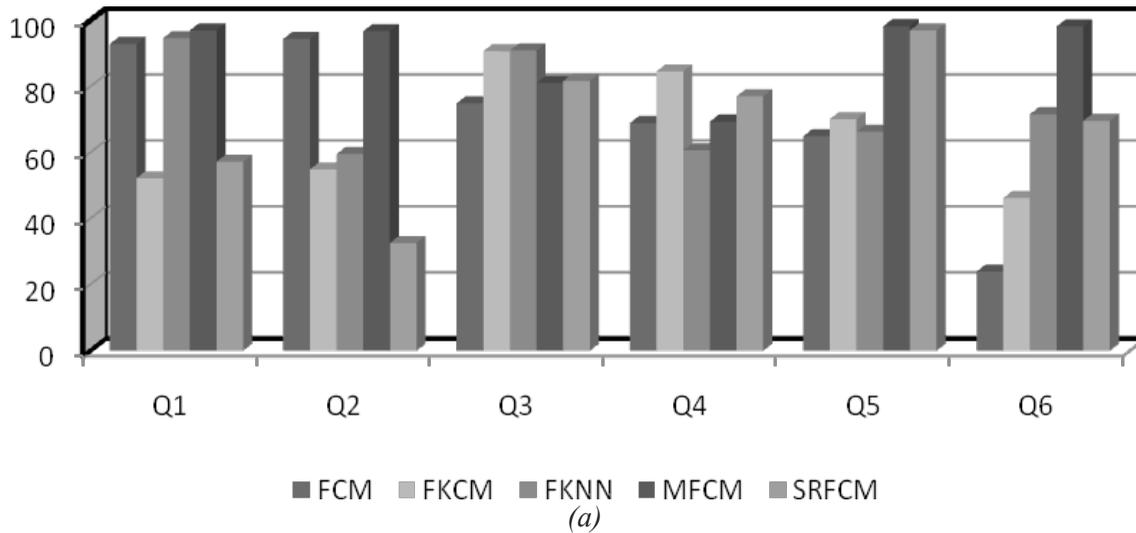
pictures the good results with FKCM are surpassed by the FKNN and MFCM algorithms. In the case of FKNN, this is due to its capacity to use several samples for each cluster, which allows a more exact calculation of the memberships and less error probability. MFCM, on the other hand, carries out a previous analysis of the histogram, which enables it in most cases to find good centroids and make good classifications.

Even though the FKNN algorithm obtains better results, in most cases it requires a high number of samples (more than 4), which may disturb the medical expert and complicate the implantation in real clini-

cal environments. This problem does not apply to the MFCM algorithm, which calculates the samples itself; however, its success values greatly vary, and for many images we had to finetune the parameters in order to obtain good results.

FUTURE TRENDS

The field of fuzzy logic is a field that evolves continuously and is increasingly applied to industrial products.

The medical images analysis field is among the most active in computerized vision and represents an important challenge to researchers in search of new technological developments.

Fuzzy clustering algorithms constitute one of the most useful and interesting branches of fuzzy logic. Their use is expected to increase and new algorithms will appear that will provide ever better results. These algorithms will more and more often be applied to the field of medical images, where they allow us to handle new multidimensional modalities and improvements.

CONCLUSION

This article presents the results obtained by various fuzzy clustering algorithms in analyzing a set of burn wound pictures. The studied techniques obtain a high level of detection in the burned area and as such show their capacity to analyse this type of medical images. Testing however reveals a high degree of variation in the values provided by each algorithm, due to the absence of homogeneous conditions during the image acquisition and the ensuing differences in the quality of the pictures.

This study shows how the FKCM algorithm provides the best results with the smallest amount of parameters. However, if we could control the context in which the photographs are taken, the best algorithm would be MFCM, which provides better results and operates automatically.

Also, we revise the state of the art in the field of fuzzy logic and clustering algorithms, in order to show the characteristics of these techniques and their possibilities.

REFERENCES

Zadeh, L. (1965). Fuzzy sets. *Information and Control*. (8) 338-353.

Shaw, I. (1998). *Fuzzy Control of Industrial Systems: Theory and Applications*. Kluwer Academic Publishers.

Yasunobu, S. & Miyamoto, S. (1985) *Automatic train operation by fuzzy predictive control*. Industrial Applications of Fuzzy Control. Ed: M. Sugeno. North Holland.

Bezdek, J., Keller, J., Krishnapuram, R., & Pal, N. (1999). *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Kluwer, Norwell, MA.

Sutton, M., Bezdek, J., Cahoon, T. (2000) *Image Segmentation by Fuzzy Clustering: Methods and Issues*. Handbook of Medical Imaging: Processing and Analysis. Ed. Isaac N. Bankman. 87-126.

Bezdek, J. (1973). *Fuzzy Mathemathics in Pattern Classification*. Ph.D. Distertation. Appl. Math., Cornell University, Ithaca, NY, 1973.

Zhong, W.D., Wei, X.X., & Jian, Y.P. (2003). Fuzzy C-Means clustering algorithm based on kernel method. *Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '03)*. IEEE Press.

Wu, C., Agam, G., Roy, A.S. & Armato, S.G. (2004). Regulated morphology approach to fuzzy shape analysis with application to blood vessel extraction in thoracic CT scans. *Proceedings of SPIE*. (5370) 1262-1270.

Vermandel, M., Betrouni, N., Taschner, C., Vasseu, C. & Rosseau, J. (2007). From MIP image to MRA segmentation using fuzzy set theory. *Computerized Medical Imaging & Graphics*. (31) 128-140.

Haußecker, H. & Tizhoosh, H.R. (1999). *Fuzzy Image Processing*. Handbook of Computer Vision and Applications. Volume 2. Ed. Bernd Jäne, Horst Haußecker and Peter Geißler. 683-727.

Givens Jr., J.A., Gray, M.R. & Keller, J.M. (1992) *A fuzzy k-nearest neighbour algorithm*. Fuzzy models for pattern recognition: methods that search for struc-

tures in data. Ed: J.C. Bezdek, S.K. Pal. IEEE Press. 258-263.

Lee, S.U. & Lim, Y.M. (1990) On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition.* (23) 9, 935-952.

Pham, D.L. (2001) Spatial models for fuzzy clustering, *Computer Vision and Image Understanding* (84) 285-297.

Gonzalez, R., Woods, R. (1996) Digital image processing. Addison-Wesley.

Zhang, Y.J. (1996) A survey on evaluation methods for image segmentation, *Pattern Recognition* (29) 1335-1346.

KEY TERMS

Fuzzification: The process of decomposing a system input and/or output into one or more fuzzy sets. Many types of curves can be used, but triangular or trapezoidal shaped membership functions are the most common.

Fuzzy Algorithm: An ordered sequence of instructions which may contain fuzzy assignments, conditional statements, repetitive statements, and traditional operations.

Fuzzy Inference Systems: A sequence of fuzzy conditional statements which may contain fuzzy assignment and conditional statements. The execution of such instructions is governed by the compositional rule of inference and the rule of preponderant alternative.

Fuzzy Operator: Operations that enable us to combine fuzzy sets. A fuzzy operator combines two fuzzy sets to give a new fuzzy set. The most frequently used fuzzy operators are the following: equality, containment, complement, intersection and union.

Medical Image: A medical specialty that uses x-rays, gamma rays, high-frequency sound waves, and magnetic fields to produce images of organs and other internal structures of the body. In diagnostic radiology the purpose is to detect and diagnose disease, whereas in interventional radiology, imaging procedures are combined with other techniques to treat certain diseases and abnormalities.

Membership Function: Gives the grade, or degree, of membership within the fuzzy set, of any element of the universe of discourse. The membership function maps the elements of the universe onto numerical values in the interval [0, 1].

Segmentation: A process that partitions a digital image into disjoint (non-overlapping) regions, using a set of features or characteristics. The output of the segmentation step is usually a set of classified elements, such as tissue regions or tissue edges.

Fuzzy Logic Estimator for Variant SNR Environments

Rosa Maria Alsina Pagès
Universitat Ramon Llull, Spain

Clàudia Mateo Segura
Universitat Ramon Llull, Spain

Joan-Claudi Socoró Carrié
Universitat Ramon Llull, Spain

INTRODUCTION

The acquisition system is one of the most sensitive stages in a Direct Sequence Spread Spectrum (DS-SS) receiver (Peterson, Ziemer & Borth, 1995), due to its critical position in order to demodulate the received information. There are several schemes to deal with this problem, such as serial search and parallel algorithms (Proakis, 1995). Serial search algorithms have slow convergence time but their computational load is very low; on the other hand, parallel systems converge very quickly but their computational load is very high. In our system, the acquisition scheme used is the multiresolutive structure presented in (Moran, Socoró, Jové, Pijoan & Tarrés, 2001), which combines quick convergence and low computational load.

The decisional system that evaluates the acquisition stage is a key process in the overall system performance, being a drawback of the structure. This becomes more important when dealing with time-varying channels, where signal to noise ratio (called SNR) is not a constant parameter. Several factors contribute to the performance of the acquistion system (Glisic & Vucetic, 1997): channel distorsion and variations, noise and interference, uncertainty about the code phase, and data randomness. The existence of all these variables led us to think about the possibility of using fuzzy logic to solve this complex acquisition estimation (Zadeh, 1973). A fuzzy logic acquisition estimator had already been tested and used in our research group to control a serial search algorithm (Alsina, Morán & Socoró, 2005) with encouraging results, and afterwards in the multiresolutive scheme (Alsina, Mateo & Socoró, 2007), and other applications to this field can be found in bibliography as (Bas, Pérez & Lagunas, 2001) or (Jang,

Ha, Seo, Lee & Lee, 1998). Several previous works have been focused in the development of acquisition systems for non frequency selective channels with fast SNR variations (Moran, Socoró, Jové, Pijoan & Tarrés, 2001) (Mateo & Alsina, 2004).

BACKGROUND

In 1964, Dr. Lofti Zadeh came out with the term *fuzzy logic* (Zadeh, 1965). The reason was that traditional logic could not answer to some questions with a simple *yes* or *no*. So, it handles the concept of partial truth. Fuzzy logic is one of the possibilities to imitate the working of a human brain, and so to try to turn artificial intelligence into real intelligence. Zadeh devised the technique as a method to solve problems for soft sciences, in particular those that involve human interaction.

Fuzzy logic has been proved to be a good option for control in very complex processes, when it is not possible to produce a mathematical model. Also fuzzy logic is recommendable for highly non-linear processes, and overall, when expert knowledge is desirable to be performed. But it is not a good idea to apply if traditional control or estimators give out satisfying results, or for problems that can be modelled in a mathematical way.

The most recent works in control and estimation using fuzzy logic applied to direct sequence spread spectrum communication systems are classified into three types. The first group uses fuzzy logic to improve the detection stage of the DS-CDMA[1] receiver, and they are presented by Bas et al and Jang et al (Bas, Pérez, & Lagunas, 2001)(Jang, Ha, Seo, Lee, & Lee, 1998). The second group uses fuzzy logic to improve interference

rejection, with works presented by Bas et al and by Chia-Chang et al (Bas, & Neira, 2003) (Chia-Chang, Hsuan-Yu, Yu-Fan, & Jyh-Horng, 2005). Finally, fuzzy logic techniques are also improving estimation and control in the acquisition stage of the DS-CDMA receiver, in works by Alsina et al (Alsina, Moran, & Socoró, 2005) (Alsina, Mateo, & Socoró, 2007).

ACQUISITION ESTIMATION IN DS-CDMA ENVIRONMENTS

One of the most important problems to be solved in direct sequence spread spectrum systems is to achieve a robust and precise acquisition of the pseudonoise sequence; this is to obtain an accurate estimation of its exact phase or timing position (Proakis, 1995). In time-varying environments this fact becomes even more important because acquisition and tracking performance can heavily degrade communication demodulation reliability. In this work a new multiresolutive acquisition system with a fuzzy logic estimator is proposed (Alsina, Mateo, & Socoró, 2007). The fuzzy logic estimation improves the accuracy of the acquisition stage compared to the results for the stability controller, through the estimation of the probability of being acquired, and the signal to noise ratio in the channel, improving the results obtained for the first fuzzy logic estimator for the multiresolutive structure in (Alsina, Mateo & Socoró, 2007).

Figure 1. Multiresolutive adaptive structure for acquisition and tracking
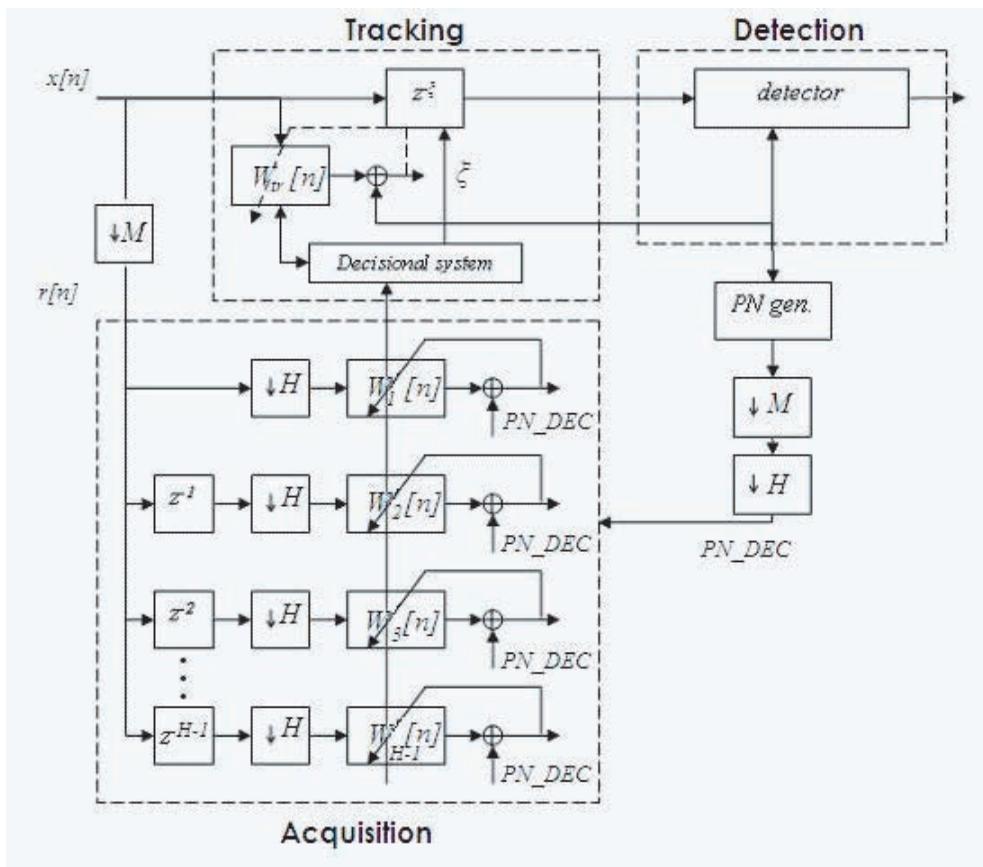
Multiresolutive Acquisition Structure

The aim of the multiresolutive scheme presented in (Moran, Socoró, Jové, Pijoan & Tarrés, 2001) is to find the correct acquisition point in a reasonable convergence time. It gives a good trade-off between speed of convergence of the parallel systems and the low computational load of the serial search algorithms. An M order decimation is firstly applied to the input signal $x[n]^2$ as acquisition stage can accept uncertainties under the chip period, and thus to decrease the computational load of the acquisition stage. Once the signal $x[n]$ is decimated, the resulting signal $r[n]$ is fed into the filters of a multiresolutive structure (see the structure in figure 1). Note that there are H different branches that work with decimated versions of the input signal, separated in H disjoint subspaces. Each branch has an adaptive FIR LMS filter of length

$$N = \left\lceil \frac{PG}{H} \right\rceil_3,$$

trained with a decimated version of the PN sequence (PN-DEC).

Under ideal conditions, in a non-frequency selective channel with white Gaussian noise, just one of the filters should locally converge an impulse like $\lambda b_i[k]\delta[n-\tau]$, where $b[k]$ is the information bit, τ represents the delay between the input signal PN sequence and the reference one and λ is the fading coefficient for channel distorsion. The algorithm is reseted every new data symbol, and a modulus smoothing average algorithm is applied to each of the LMS solutions ($w_i[n]$) to remove the data randomness component $b_i[k]$ dependency, obtaining nonnegative and averaged impulsional responses ($Wav_i[n]$). The decisional system uses a peak detection algorithm to find which of these filters has detected the signal ($W_{con}[n]$), and the position of the maximum (τ) in this filter will give the coarse estimation of the acquisition phase.

When the acquisition point by the decisional system is restored, tracking is solved with another adaptive LMS filter ($w_r[n]$), which expands the search window around the acquisition point, using the full time resolution input signal $x[n]$. Thus, the estimation of the acquisition point (now called ξ) is refined by the tracking and the signal can be correctly demodulated.

The Fuzzy Logic Acquisition Estimation

The fuzzy logic acquisition estimator has been designed using data of the impulsional response of all the LMS filters of the structure. Their values variations give information about the probability of being correctly acquired, and also about SNR ratio variations in the channel. In the conducted experiments, the signal space has been divided into four subspaces ($H=4$), so four LMS filters compose the acquisition stage. The length of the PN sequences is $PG=127$, so each filter has

$$N = \left\lceil \frac{PG}{H} \right\rceil = 32$$

taps to converge. This input and output variables were already defined in (Alsina, Mateo & Socoró, 2007), but the rules to be evaluated have been designed in a more precise way.

Input Variables

Four different parameters have been defined as inputs in the fuzzy estimator; three of them referred to the values of the four modulus averaged acquisition LMS filters ($Wav_i[n]$), especially the LMS filter adapted to the decimated sequence PN-DEC (called $W_{con}[n]$), and one about the tracking filter ($w_{tr}[n]$) that refines the search:

*   *Ratio₁*: it is computed as the quotient of the peak value of the LMS filter $W_{con}[n]$ divided into the mean value of this filter but the maximum, as follows:

$$Ratio_1 = \frac{W_{con}[\tau]}{\frac{1}{N}\sum_{n=1;\, n\neq\tau}^{N} W_{con}[n]}$$

*   *Ratio₂*: it is evaluated as the quotient of the peak value of the LMS filter $W_{con}[\tau]$ divided into the average of the value of the same position in the other three filters $Wav_i[n]$.

$$Ratio_2 = \frac{W_{con}[\tau]}{\frac{1}{H-1}\sum_{i=1;\, Wav_i\neq W_{con}}^{H} Wav_i[\tau]}$$

- *Ratio$_3$*: it is obtained as the quotient of the peak value of the LMS filter $W_{con}[\tau]$ divided into the mean value of the three other filters $Wav_i[n]$.

$$Ratio_3 = \frac{W_{con}[\tau]}{\frac{1}{H-1}\sum_{i=1;\,Wav_i \neq W_{con}}^{H}\frac{1}{N}\sum_{n=1}^{N}Wav_i[n]}$$

- *Ratio$_{1\,track}$*: it is computed as the quotient of the peak value of the LMS tracking filter $w_{tr}[\xi]$, being ξ the most precise estimation of the correct acquisition point, divided into the mean value of the same filter but the maximum.

$$Ratio_{1\_track} = \frac{w_{tr}[\xi]}{\frac{1}{N}\sum_{n=1;\,n\neq\xi}^{N}w_{tr}[n]}$$

These parameters have been chosen due to the information they contain about the probability of being acquired, and also about the SNR level in the channel and its variations. They value variations give good estimations about acquisition quality and a good measure for SNR, with the appropriate definition of IF-THEN rules.

Output Variables

The results will be obtained using a defuzzyfication method based on the centroid (Leekwijck & Kerre,

Figure 2. Variable acquisition for all input variables combinations
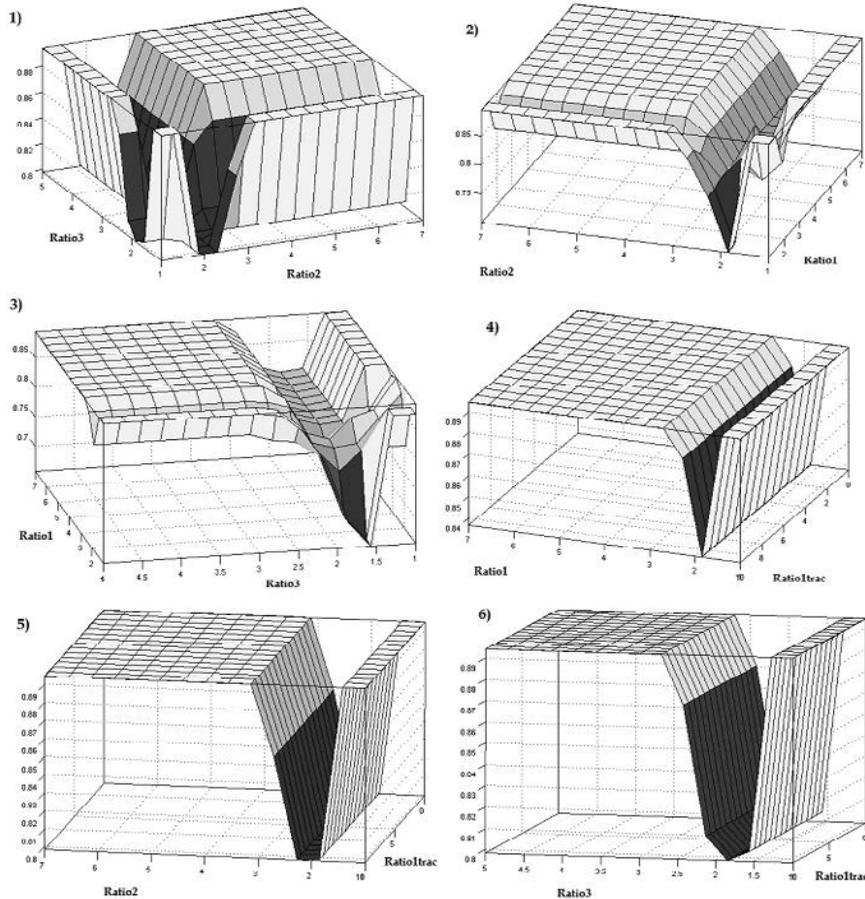
1999). Two output variables will be computed. *Acquisition*, giving a value in the range of *[0,1]*, being zero when it is *Not Acquired* and one if it is *Acquired*. Three more fuzzy sets have been defined between the extreme values; *Probably Not Acquired*, *Not Determined* and *Probably Acquired*. *Acquisition* will show a value of reliability for the correct demodulation of the detector. The multiresolutive scheme only gives an estimation of the acquisition point, and *Acquisition* value evaluates the probability of being acquired, and so, the consistency of the bit demodulation done by the receiver.

The second variable is *SNR Estimation* which gives a value (in the range of *[-30,0] dBs* in our experiment) of the estimated SNR value in the channel. *SNR Estimation* will give us information about channel conditions; this will help not only in acquisition and tracking, but also in detection as in (Verdú, 1998) or (Alsina, Morán & Socoró, 2005).

If-Then Rules

A total of sixty rules have been used to define the two outputs in function of the input values, evolving the set of rules used in (Alsina, Mateo & Socoró). In figure 2 the surface for *Acquisition* for all input variables and figure 3 shows the surface for *SNR Estimation* for all inputs. Rules have been defined to take into account the best performance, in its range, of each input parameter value to design the two outputs of the fuzzy estimator. This means the value range is only considered where their estimations are more reliable for both outputs.

Figure 3. Variable SNR Estimation for all input variables combinations
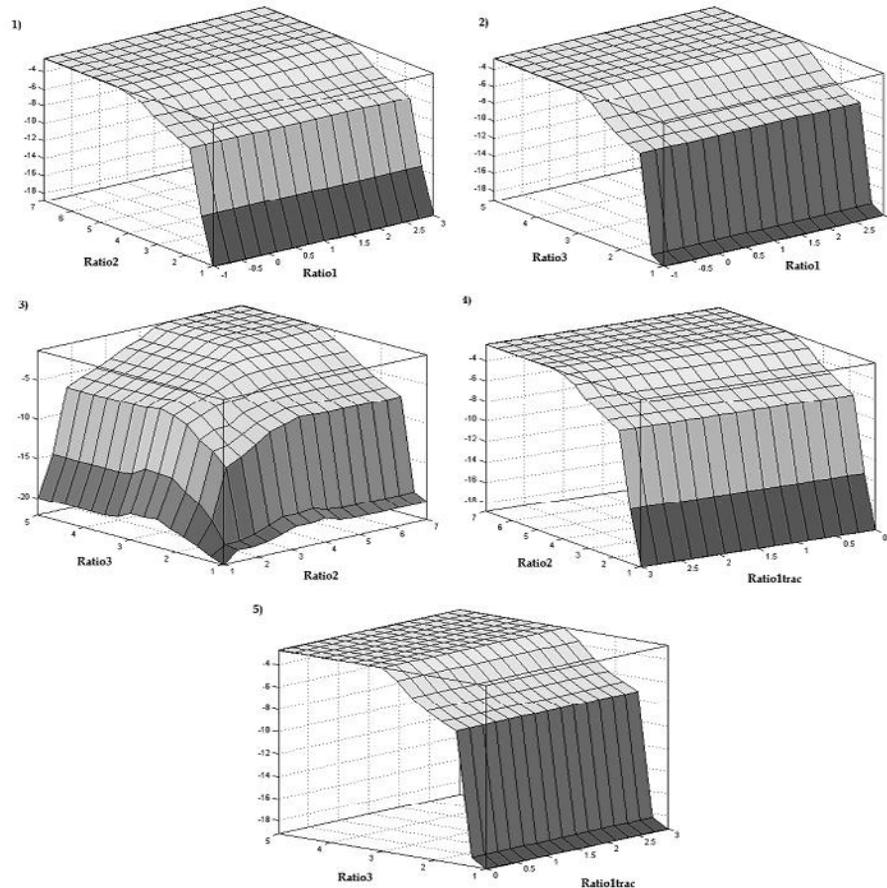
The most improved estimation for the output *Acquisition* is the correspondence to *Not Determined*; this means that the input parameters have no coherent values of *Acquisition* or *Not Acquisition* by themselves. To obtain a precise output value, the fuzzy estimator evaluates the degree of implication of each input parameter to the membership functions, and projects this implication to the fuzzy sets of the output variable *Acquisition*, in order to obtain its value through defuzzyfication. $Ratio_1$ and $Ratio_{1\,track}$ are the best input parameters to estimate *Acquisition* when channel conditions are good; these two parameters are supported by $Ratio_2$ and $Ratio_3$ when SNR worsen. The precision of the critical estimations has been improved in the design of the new rules for the fuzzy estimator.

On the other hand, *SNR Estimation* most robust evaluations are made by $Ratio_2$ and $Ratio_3$; they are improved by $Ratio_{1\,track}$ when SNR is high, and by $Ratio_1$ when SNR is very low. As can be observed in figure 3, these variables highly correlate with *SNR Estimation* value.

Results

In this section the results obtained with the new acquisition and SNR fuzzy logic estimator will be summarized. Several simulations using an Additive White Gaussian Noise channel (AWGN), some of them with very fast SNR changes, have been done to show the performance of the fuzzy estimator in terms of reliability and stability.

Fuzzy Estimator Acquisition Reliability vs. Stability Control

A previous acquisition estimation was obtained using a *stability* control (Moran, Socoró, Jové, Pijoan & Tarrés, 2001), that took into account preservation of the acquisition point for evaluation and comparison purposes. It considered that the system was acquired only due to continuous repetitions of the acquisition point given by the multiresolutive scheme. This *stability* control gave a binary response about the performance

Figure 4. % of correct estimation of acquisition using the new fuzzy estimator against the stability Control

F
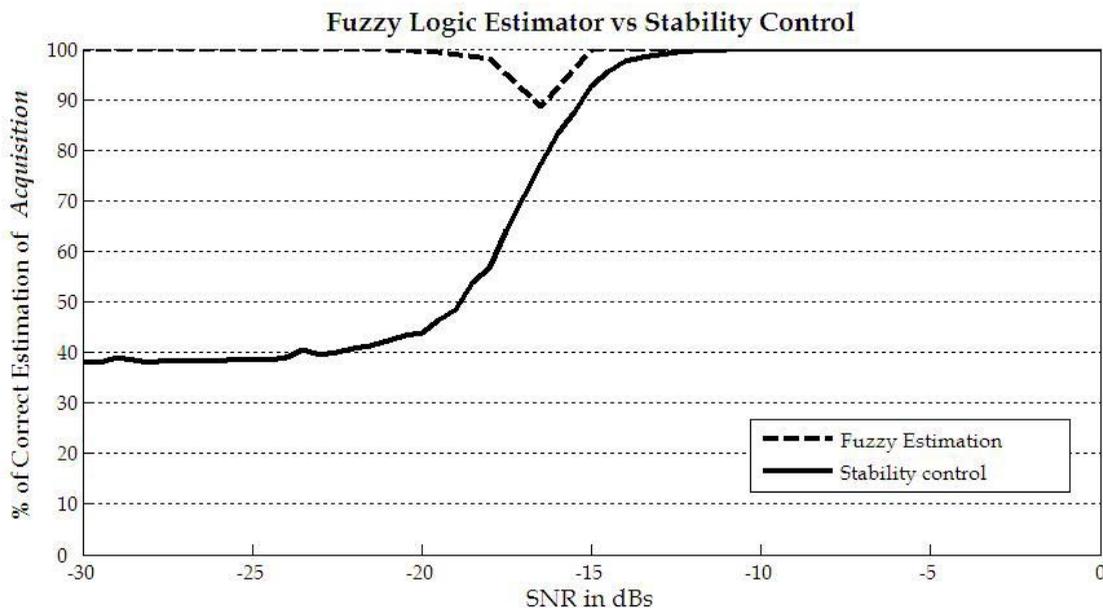
of the system. Despite its good performance, being observed in figure 4, the new fuzzy approach improves the results for wider SNR range. The quality of the fuzzy acquisition estimation is much better for very low SNR compared to the *stability* control, and its global performance for the whole range of SNR in our tests is improved. The *stability* control is not a good estimator for critical SNR (considered around *-15dBs*), and it decreases its reliability when SNR decreases. Despite showing similar performance around critical SNR, the fuzzy logic estimation of *Acquisition* improves its performance for worse SNR ratios, being over 90% of correct estimation all the simulations along.

Fuzzy SNR Estimation in Time Varying Channels

In figure 5.a the acquisition system has been simulated in an AWGN channel, forcing severe and very fast SNR changes in order to evaluate the convergence speed of the SNR estimator. *SNR Estimation* mean value, being a very variable value, is obtained through an exponential smoothing average filter, and compared to the SNR in the AWGN channel. The SNR in the channel is estimated

quite precisely until very low SNR (near *-20dBs*) by the fuzzy block, as the input parameters are not stable enough to make a good prediction for lower values; this is similar to what happens for *Acquisition* estimation. To observe the recovery of the fuzzy estimator in case of fast SNR changes in the channel, a detail of *SNR Estimation* is shown in figure 5.b. This information shows the channel state to the receiver, and allows further work to improve reliability of the demodulation by means of different approaches (Verdú, 1998).

FUTURE TRENDS

Future work will be focused on improving the estimation for the SNR in the fuzzy system. Another goal to be reached is to increase the stability against channel changes using previous detected symbols, obtaining a system with feedback. The fuzzy estimator outputs will be used to design a controller for the acquisition and tracking structure. Its aim will be to improve the stability of estimation of the correct acquisition point (ξ) through an effective and robust control of its variations for sudden channel changes, so memory will be added to the fuzzy logic estimator. This way the estimator is

Figure 5. a) SNR estimation in a varying SNR channel; b) Detail of SNR Estimation when adapting to an instantaneous SNR variation

725
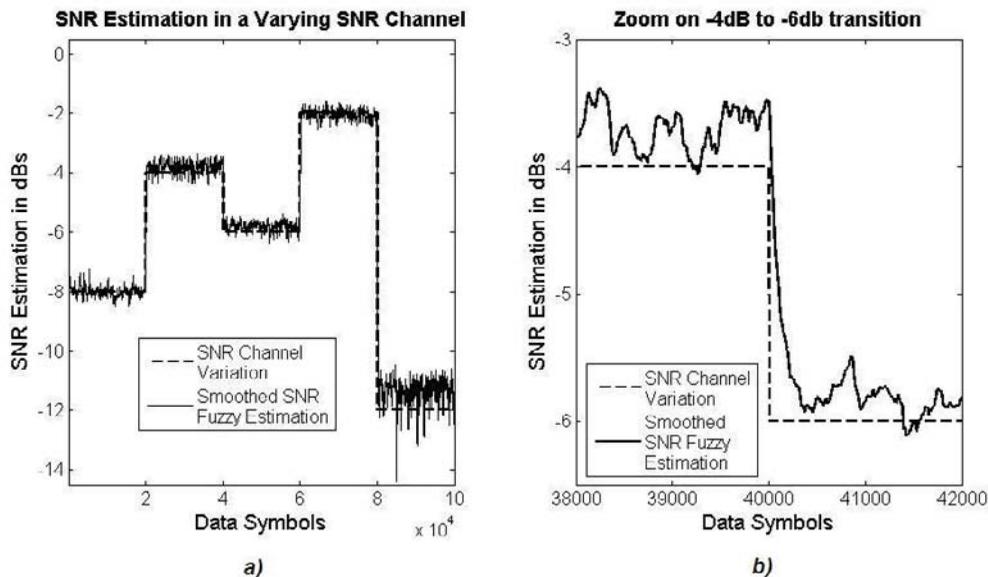
converted in a controller, and the whole performance of the receiver is improved.

Further research will also take into account multipath channel conditions and possible variations, including rake-based receiver detection, in order to reach a good acquisition and tracking performance in ionospheric channels. Furthermore, the reliability of the results encourages us to use the acquisition estimation to minimize the computational load of the acquisition system for proper channel conditions, thorough decreasing the number of iterations to converge in the LMS adaptive filters. A more efficient fuzzy logic control can be designed in order to achieve a better trade-off between computational load (referred to the LMS filters adaptation) and acquisition point estimation accuracy (ξ).

CONCLUSION

The new proposed acquisition system estimator has already been exposed, and some results have been compared against a *stability* control strategy within the multiresolutive acquisition system in a variant SNR environment. The main advantage of a multiresolutive fuzzy estimator is its reliability when evaluating the probability of acquisition, also its stability, and its quick convergence when there are fast channel SNR changes. The computational load of a fuzzy estimator is higher than the same cost in a *stability* control. The mean number of FLOPS in a DSP needed to do all the process is greater compared to the conventional stability control. This has to be taken into account because the multiresolutive structure should make its computational cost minimum to work on-line with the received data. Further work will be done to compare the computational load added to the structure to the global improvements of the multiresolutive receiver, to decide whether this cost increase is affordable for the acquisition system, or it is not.

REFERENCES

Alsina, R.M., Morán, J.A., & Socoró, J.C. (2003). Multiresolution Adaptive Structure for Acquisition and Detection in DS-SS Digital Receiver in a Multiuser Environment. *IEEE International Symposium on Signal Processing and its Applications.*

Alsina, R.M., Morán, J.A., & Socoró, J.C. (2005). Sequential PN Acquisition Based on a Fuzzy Logic Controller. *8th International Workshop on Artificial Neural Networks, Lecture Notes in Computer Science.* (3512) 1238-1245.

Alsina, R.M., Mateo, C., & Socoró, J.C. (2007). Multiresolutive Adaptive PN Acquisition Scheme with a Fuzzy Logic Estimator in Non Selective Fast SNR Variation Environments. *9th International Workshop on Artificial Neural Networks, Lecture Notes in Computer Science.* (4507) 367-374.

Bas, J., Pérez, A., & Lagunas, M.A. (2001). Fuzzy Recursive Symbol-by-Symbol Detector for Single User CDMA Receivers. *International Conference on Acoustics, Speech and Signal Processing.*

Bas, J., & Neira, A.P. (2003). A fuzzy logic system for interference rejection in code division multiple access. *The 12th IEEE International Conference on Fuzzy Systems,* (2), 996-1001.

Chia-Chang, H., Hsuan-Yu, L., Yu-Fan, C., & Jyh-Horng, W. (2005). Adaptive interference supression using fuzzy-logic-based space-time filtering techniques in multipath DS-CDMA. *The 6th IEEE International Workshop on Signal Processing Advances in Wireless Communications*, p. 22-26.

Glisic, S.G., & Vucetic, B. (1997). *Spread Spectrum CDMA Systems for Wireless Communications*. Artech House Publishers.

Jang, J., Ha, K., Seo, B., Lee, S., & Lee, C.W. (1998). A Fuzzy Adaptive Multiuser Detector in CDMA Communication Systems. *International Conference on Communications.*

Leekwijck, W.V., & Kerre, E.E. (1999). Defuzzification: Criteria and Classification. *Fuzzy Sets and Systems.* (108) 159-178.

Mateo, C., & Alsina, R.M. (2004). Diseno de un Sistema de Control Adaptativo a las Condiciones del Canal para un Sistema de Adquisición de un Receptor DS-SS. *XIX Congreso de la Unión Científica Nacional de Radio.*

Morán, J.A., Socoró, J.C., Jové, X., Pijoan, J.L., & Tarrés, F. (2001). Multiresolution Adaptive Structure for Acquisition in DS-SS Receiver. *International Conference on Acoustics, Speech and Signal Processing.*

F

Peterson, R.L., Ziemer, R.E., & Borth, D.E. (1995). *Spread Spectrum Communications Handbook*. Prentice Hall.

Proakis, J.G. (1995). *Digital Communications*. Mc-Graw-Hill.

Verdú, S. (1998). *Multiuser Detection*. Cambridge University Press.

Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control*. (8), 338-353.

Zadeh, L.A. (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions Systems Man Cybernetics*. (3), 28-44.

Zadeh, L.A. (1988). Fuzzy Logic. *Computer*, 83-92.

KEY TERMS

Defuzzyfication: After computing the fuzzy rules, and evaluating the fuzzy variables, this is the process the system follows to obtain a new membership function for each output variable.

Degree of Truth: It denotes the extent to which a preposition is true. It is important to not be confused with the concept of probability.

Fuzzy Logic: Fuzzy logic was derived from Fuzzy Set theory, working with a reason that it is approximate rather than precise, deducted from the typical predicate logic.

Fuzzy Sets: Fuzzy sets are sets whose members have a degree of membership. They were introduced to be an extension of the classical sets, whose elements' membership was assessed by binary numbers.

Fuzzyfication: It is the process of defining the degree of membership of a crisp value for each fuzzy set.

IF-THEN Rules: They are the typical rules used by expert fuzzy systems. The IF part is the antecedent, also named premise, and the THEN part is the conclusion.

Linguistic Variables: They take on linguistic values, which are words, with associated degrees of membership in each set.

Linguistic Term: It is a subjective category for a linguistic variable. Each linguistic term is associated with a fuzzy set.

Membership Function: It is the function that gives the subjective measures for the linguistic terms.

ENDNOTES

[1] DS-CDMA stands for Direct Sequence Code Division Multiple Access.

[2] The received signal $x[n]$ is sampled at M samples per chip in order to give the necessary time resolution for the tracking stage.

[3] where PG is the length of the pseudonoise sequences, also called PN sequences and 'ceil(x)' (expressed as N=[x]) is the smaller integer greater than x.

Fuzzy Rule Interpolation

Szilveszter Kovács
University of Miskolc, Hungary

INTRODUCTION

The "fuzzy dot" (or fuzzy relation) representation of fuzzy rules in fuzzy rule based systems, in case of classical fuzzy reasoning methods (e.g. the Zadeh-Mamdani-Larsen Compositional Rule of Inference (CRI) (Zadeh, 1973) (Mamdani, 1975) (Larsen, 1980) or the Takagi - Sugeno fuzzy inference (Sugeno, 1985) (Takagi & Sugeno, 1985)), are assuming the completeness of the fuzzy rule base. If there are some rules missing i.e. the rule base is "sparse", observations may exist which hit no rule in the rule base and therefore no conclusion can be obtained. One way of handling the "fuzzy dot" knowledge representation in case of sparse fuzzy rule bases is the application of the Fuzzy Rule Interpolation (FRI) methods, where the derivable rules are deliberately missing. Since FRI methods can provide reasonable (interpolated) conclusions even if none of the existing rules fires under the current observation. From the beginning of 1990s numerous FRI methods have been proposed. The main goal of this article is to give a brief but comprehensive introduction to the existing FRI methods.

BACKGROUND

Since the classical fuzzy reasoning methods (e.g. the Zadeh-Mamdani-Larsen CRI) are demanding complete rule bases, the classical rule base construction claims a special care of filling all the possible rules. In case if the rule base is "sparse" (some rules are missing), observations may exist which hit no rule and hence no conclusion can be obtained. In many application areas of fuzzy control structures, the accidental lack of conclusion is hard to explain, or meaningless (e.g. in steering control of a vehicle). This case one obvious solution could be to keep the last real conclusion instead of the missing one, but applying historical data automatically to fill undeliberately missing rules could cause unpredictable side effects. Another solution for the same problem is the application of the fuzzy rule interpolation (FRI) methods, where the derivable rules are deliberately missing. The rule base of an FRI controller is not necessarily complete, since FRI methods can provide reasonable (interpolated) conclusions even if none of the existing rules fires under the current observation. It could contain the most significant fuzzy rules only, without risking the chance of having no conclusion for some of the observations. On the other hand most of the FRI methods are sharing the burden of high computational demand, e.g. the task of searching for the two closest surrounding rules to the observation, and calculating the conclusion at least in some characteristic α-cuts. Moreover in some methods the interpretability of the fuzzy conclusion gained is also not straightforward (Kóczy & Kovács, 1993). There have been a lot of efforts to rectify the interpretability of the interpolated fuzzy conclusion (Tikk & Baranyi, 2000). In (Baranyi, Kóczy & Gedeon, 2004) Baranyi *et al.* give a comprehensive overview of the recent existing FRI methods. Beyond these problems, some of the FRI methods are originally defined for one dimensional input space, and need special extension for the multidimensional case (e.g. (Jenei, 2001), (Jenei, Klement & Konzel, 2002)). In (Wong, Tikk, Gedeon & Kóczy, 2005) Wong *et al.* gave a comparative overview of the recent multidimensional input space capable FRI methods. In (Jenei, 2001) Jenei introduced a way for axiomatic treatment of the FRI methods. In (Perfilieva, 2004) Perfilieva studies the solvability of fuzzy relation equations as the solvability of interpolating and approximating fuzzy functions with respect to a given set of fuzzy rules (e.g. fuzzy data as ordered pairs of fuzzy sets). The high computational demand, mainly the search for the two closest surrounding rules to an arbitrary observation in the multidimensional antecedent space turns many of these methods hardly suitable for real-time applications. Some FRI methods, e.g. the method introduced by Jenei *et al.* in (Jenei, Klement & Konzel, 2002), eliminate the search for the two closest surrounding rules by taking all the rules into consideration, and therefore speeding up the reasoning process. On the other hand, keeping the goal of con-

F

structing fuzzy conclusion, and not simply speeding up the reasoning, they still require some additional (or repeated) computational steps for the elements of the level set (or at least for some relevant α levels). An application oriented aspect of the FRI emerges in (Kovács, 2006), where for the sake of reasoning speed and direct real-time applicability, the fuzziness of fuzzy partitions replaced by the concept of Vague Environment (Klawonn, 1994).

In the followings, the brief structure of several FRI methods will be introduced in more details.

FUZZY RULE INTERPOLATION METHODS

One of the first FRI techniques was published by Kóczy and Hirota (Kóczy & Hirota, 1991). It is usually referred as KH method. It is applicable to convex and normal fuzzy (CNF) sets in single input and single output (SISO) systems. The KH method takes into consideration only the two closest surrounding (flanking) rules to the observation. It determines the conclusion by its α-cuts in such a way that the ratio of distances between the conclusion and the consequents should be identical with the ratio of distances between the observation and the antecedents for all important α-cuts. The applied formula:

$$d\left(A^*, A_1\right) : d\left(A^*, A_2\right) = d\left(B^*, B_1\right) : d\left(B^*, B_2\right),$$

can be solved for the required conclusion B^* for relevant α-cuts after decomposition. Where $A_1 \rightarrow B_1$ and $A_2 \rightarrow B_2$ are the two flanking rules of the observation A^* and d: $F(X) \times F(X) \rightarrow R$ is a distance function of fuzzy sets (in case of the KH method it was calculated as the distance of the lower and upper end points of the α-cuts) (see e.g. on Fig. 1.).

It is shown in, e.g. in (Kóczy & Kovács, 1993), (Kóczy & Kovács, 1994) that the conclusion of the KH method is not always directly interpretable as fuzzy set (see e.g. on Fig. 1.). This drawback motivated many alternative solutions. The first modification was proposed by Vass, Kalmár and Kóczy (Vass, Kalmár & Kóczy, 1992) (referred as VKK method), where the conclusion is computed based on the distance of the centre points and the widths of the α-cuts, instead of their lower and upper end point distances. The VKK method extends the applicability of the KH method, but it was still strongly depends on the membership shape of the fuzzy sets (e.g. it was unable to handle singleton antecedent sets, as the width of the antecedent's support must not be zero).

In spite of the known restrictions, the KH method is still popular because of its simplicity. Subsequently it was generalized in several ways. Among them the stabilized KH interpolator was emerged, as it was proved

Figure 1. KH method for two SISO rules: $A_1 \rightarrow B_1$ and $A_2 \rightarrow B_2$, conclusion y of the observation x
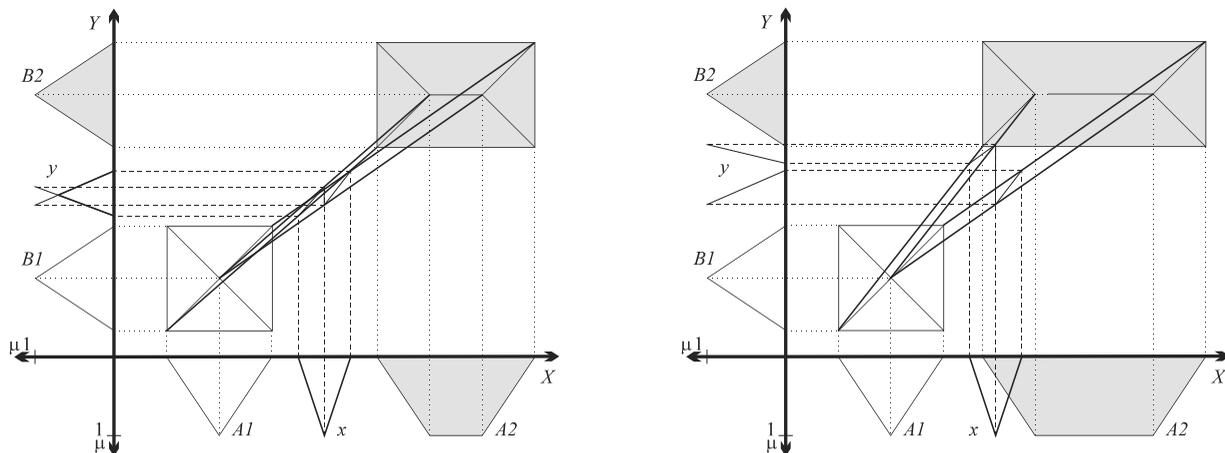
to hold the universal approximation property in (Tikk, Joó, Kóczy, Várlaki, Moser & Gedeon, 2002) and (Tikk, 2003). This method takes into account all the rules of the rule base in the calculation of the conclusion. The method adapts a modification of the *Shepard operator* based interpolation (Shepard, 1968). The rules are taken into account in extent to the inverse of the distance between their antecedents and the observation. The universal approximation property holds if the distance function is raised to the power of at least the number of the antecedent dimension.

Another modification of KH method is the modified alpha-cut based interpolation method (referred as MACI) (fully extended in (Tikk & Baranyi, 2000)), which alleviates completely the abnormality problem. MACI's main idea is the following: it transforms fuzzy sets of the input and output universes to such a space where abnormality is excluded, then computes the conclusion there, which is finally transformed back to the original space. MACI uses vector representation of fuzzy sets. The original method was introduced in (Yam & Kóczy, 1997) and it was applicable for CNF sets only. This restriction was latter relaxed in (Tikk, Baranyi, Gedeon & Muresan 2001) by paying its expanse in higher computational demand than the original method. MACI is one of the most applied FRI methods (Wong, Tikk, Gedeon & Kóczy, 2005), since it preserves advantageous computational and approximate nature of KH method, while it excludes its chance for abnormal conclusion.

Another FRI method was proposed by Kóczy *et al.* in (Kóczy, Hirota & Gedeon, 1997). It takes into consideration only the two closest surrounding rules to the observation and its main idea is the conservation of the "relative fuzziness" (referred as CRF method). This notion means that the left (and right) fuzziness of the approximated conclusion in proportion to the flanking fuzziness of the neighbouring consequent should be the same as the left (and right) fuzziness of the observation in proportion to the flanking fuzziness of the neighbouring antecedent. The original method is restricted to CNF sets only.

An improved fuzzy interpolation technique for multidimensional input spaces (referred as IMUL) was originally proposed in (Wong, Gedeon & Tikk, 2000), and described more detailed in (Wong, Tikk, Gedeon & Kóczy, 2005). IMUL applies a combination of CRF and MACI methods, and mixes the advantages of both. The core of the conclusion is determined by MACI method,

while its flanks by CRF (the method is restricted to trapezoidal membership functions). The main advantages of this method are its applicability for multi-dimensional problems and its relative simplicity.

Conceptually different approaches were proposed in (Baranyi, Kóczy & Gedeon, 2004) based on the relation, semantic and inter-relational features of the fuzzy sets. The family of these methods applies a two step "General Methodology" (referred as GM). The notation also reflects the feature, that methods based on GM can handle arbitrary shaped fuzzy sets. The basic concept is to divide the task of the FRI into two main steps. The first step is to determine the reference point of the conclusion based on the ratio of the distances between the reference points of the observation and the antecedents. Then accomplishing the first step, based on the existing rules a new, interpolated rule is generated for the reference point of the observation and the reference point of the conclusion. In the second step of the method, a single rule reasoning method (revision function) is applied to determine the final fuzzy conclusion based on the similarity of the fuzzy observation and the antecedent of the new "interpolated" rule. For both the main steps of GM numerous solutions exists, therefore the GM stands for an FRI concept, or a family of FRI methods.

A rather different application oriented aspect of the FRI emerges in the concept of the Fuzzy Interpolation based on Vague Environment FRI method (referred as FIVE), originally introduced in (Kovács, 1996), (Kovács & Kóczy, 1997a), (Kovács & Kóczy, 1997b) and extended with the ability of handling fuzzy observation in (Kovács, 2006). It was developed to fit the speed requirements of direct fuzzy control, where the conclusions of the fuzzy controller are applied directly as control actions in a real-time system. The main idea of the FIVE method is based on the fact that most of the control applications serves crisp observations and requires crisp conclusions from the controller. Adopting the idea of the vague environment (Klawonn, 1994), FIVE can handle the antecedent and consequent fuzzy partitions of the fuzzy rule base by scaling functions (Klawonn, 1994) and therefore turn the fuzzy interpolation to crisp interpolation. In FIVE any crisp interpolation, extrapolation, or regression method can be adapted very simply for FRI. Because of its simple multidimensional applicability, in FIVE, originally the *Shepard operator* based interpolation (Shepard, 1968) was adapted.

F

FUTURE TRENDS

Future trends of the FRI methods include the appearance of numerous hybrid FRI methods i.e. neuro-FRI, genetic-FRI for (depending on the application area) gradient based, or gradient free parameter optimisation of the FRI model. Future trends also directed to extended number of practical applications of the FRI. Recently a freely available comprehensive FRI toolbox (Johanyák, Tikk, Kovács & Wong, 2006) and an FRI oriented web site (http://fri.gamf.hu) were appeared for aiding and guiding the future FRI applications.

CONCLUSION

There are relatively few Fuzzy Rule Interpolation (FRI) techniques can be found among the practical fuzzy rule based applications. On one hand the FRI methods are not widely known, and some of them have limitations from practical application point of view, e.g. can be applied only in one dimensional case, or defined based on the two closest surrounding rules of the actual observation. On the other hand enabling the application of sparse rule bases the FRI methods can dramatically simplify the way of fuzzy rule base creation, since FRI methods can provide reasonable (interpolated) conclusions even if none of the existing rules fires under the current observation. Therefore these methods can save the expert from dealing with derivable rules and help to concentrate on cardinal actions only and hence simplify the rule base creation itself. Thus, compared to the classical fuzzy CRI, the number of the fuzzy rules needed to be handled during the design process, could be dramatically reduced (see e.g. in (Kovács, 2005)). Moreover in case of parameter optimisation of the sparse FRI model (hybrid FRI methods), the reduced FRI rule base size could also means reduction in the size of the optimisation search space, and hence it can lead to quicker optimisation algorithms too.

REFERENCES

P. Baranyi, L. T. Kóczy, and T. D. Gedeon (2004). *A Generalized Concept for Fuzzy Rule Interpolation*. IEEE Transaction on Fuzzy Systems, (12) 6, 820-837.

S. Jenei (2001). *Interpolating and extrapolating fuzzy quantities revisited – an axiomatic approach*. Soft Computing, (5), 179-193.

S. Jenei, E. P. Klement and R. Konzel (2002). *Interpolation and extrapolation of fuzzy quantities – The multiple-dimensional case*. Soft Computing, (6), 258-270.

Zs. Cs. Johanyák, D. Tikk, Sz. Kovács, K. W. Wong (2006). *Fuzzy Rule Interpolation Matlab Toolbox – FRI Toolbox*, Proc. of the IEEE World Congress on Computational Intelligence (WCCI'06), 15th Int. Conf. on Fuzzy Systems (FUZZ-IEEE'06), Vancouver, BC, Canada, Omnipress. ISBN 0-7803-9489-5, 1427-1433.

F. Klawonn (1994). *Fuzzy Sets and Vague Environments*. Fuzzy Sets and Systems, (66), 207-221.

G. J. Klir, T. A. Folger (1988). *Fuzzy Sets Uncertainity and Information*. Prentice-Hall International.

L. T. Kóczy and K. Hirota (1991). *Rule interpolation by α-level sets in fuzzy approximate reasoning*. BUSEFAL, Automne, URA-CNRS, Toulouse, France, (46), 115-123.

L. T. Kóczy and Sz. Kovács (1993). *On the preservation of the convexity and piecewise linearity in linear fuzzy rule interpolation*. Tokyo Institute of Technology, Yokohama, Japan, Technical Report TR 93-94/402, LIFE Chair Fuzzy Theory.

L. T. Kóczy and Sz. Kovács (1994). *Shape of the Fuzzy Conclusion Generated by Linear Interpolation in Trapezoidal Fuzzy Rule Bases*. Proceedings of the 2nd European Congress on Intelligent Techniques and Soft Computing, Aachen, 1666–1670.

L.T. Kóczy, K. Hirota, and T. D. Gedeon (1997). *Fuzzy rule interpolation by the conservation of relative fuzziness*. Technical Report TR 97/2. Hirota Lab, Dept. of Comp. Int. and Sys. Sci., Tokyo Institute of Technology, Yokohama.

Sz. Kovács (1996). *New Aspects of Interpolative Reasoning*. Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, 477-482.

Sz. Kovács, and L.T. Kóczy (1997a). *Approximate Fuzzy Reasoning Based on Interpolation in the Vague*

Environment of the Fuzzy Rule base as a Practical Alternative of the Classical CRI. Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic, 144-149.

Sz. Kovács, and L.T. Kóczy (1997b). *The use of the concept of vague environment in approximate fuzzy reasoning*. Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, Bratislava, Slovak Republic, (12), 169-181.

Sz. Kovács (2005). *Interpolative Fuzzy Reasoning in Behaviour-based Control*, Advances in Soft Computing, Computational Intelligence, Theory and Applications, Bernd Reusch (Ed.), Springer, Germany, ISBN 3-540-22807-1, (2), 159-170.

Sz. Kovács (2006). *Extending the Fuzzy Rule Interpolation "FIVE" by Fuzzy Observation*. Advances in Soft Computing, Computational Intelligence, Theory and Applications, Bernd Reusch (Ed.), Springer Germany, ISBN 3-540-34780-1, 485-497.

P. M. Larsen (1980). *Industrial application of fuzzy logic control.* Int. J. of Man Machine Studies, (12) 4, 3-10.

E. H. Mamdani and S. Assilian (1975). *An experiment in linguistic synthesis with a fuzzy logic controller.* Int. J. of Man Machine Studies, (7), 1-13.

I. Perfilieva (2004). *Fuzzy function as an approximate solution to a system of fuzzy relation equations*. Fuzzy Sets and Systems, (147), 363-383.

D. Shepard (1968). *A two dimensional interpolation function for irregularly spaced data*. Proc. 23rd ACM Internat. Conf., 517-524.

M. Sugeno (1985). *An introductory survey of fuzzy control.* Information Science, (36), 59-83.

T. Takagi and M. Sugeno (1985). *Fuzzy identification of systems and its applications to modeling and control.* IEEE Trans. on SMC, (15), 116-132.

D. Tikk and P. Baranyi (2000). *Comprehensive analysis of a new fuzzy rule interpolation method*. In IEEE Transaction on Fuzzy Systems, (8) 3, 281-296.

D. Tikk, P. Baranyi, T. D. Gedeon, and L. Muresan (2001). *Generalization of a rule interpolation method resulting always in acceptable conclusion*. Tatra Mountains Mathematical Publications, (21), 73-91.

D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon (2002). *Stability of interpolative fuzzy KH-controllers*. Fuzzy Sets and Systems, (125) 1, 105-119.

D. Tikk (2003). *Notes on the approximation rate of fuzzy KH interpolator*. Fuzzy Sets and Systems, (138) 2, 441-453.

Y. Yam, and L. T. Kóczy (1997). *Representing membership functions as points in high dimensional spaces for fuzzy interpolation and extrapolation*. Dept. Mech. Automat. Eng., Chinese Univ. Hong Kong, Technical Report CUHK-MAE-97-03.

G. Vass, L. Kalmár and L. T. Kóczy (1992). *Extension of the fuzzy rule interpolation method*. Proceedings of the International Conference Fuzzy Sets Theory Applications (FSTA'92), Liptovsky Mikulas, Czechoslovakia, 1-6.

K. W. Wong, T. D. Gedeon, and D. Tikk (2000). *An improved multidimensional α-cut based fuzzy interpolation technique*. Proceedings of the International Conference Artificial Intelligence in Science and Technology (AISAT'2000), Hobart, Australia, 29-32.

K. W. Wong, D. Tikk, T. D. Gedeon, and L. T. Kóczy (2005). *Fuzzy Rule Interpolation for Multidimensional Input Spaces With Applications*. IEEE Transactions on Fuzzy Systems, ISSN 1063-6706, (13) 6, 809-819.

L. A. Zadeh (1973). *Outline of a new approach to the analysis of complex systems and decision processes*. IEEE Trans. on SMC, (3), 28-44.

KEY TERMS

α-Cut of a Fuzzy Set: Is a crisp set, which holds the elements of a fuzzy set (on the same universe of discourse) whose membership grade is grater than, or equal to α. (In case of "strong" α -cut it must be grater than α.)

ε-Covering Fuzzy Partition: The fuzzy partition (a set of linguistic terms (fuzzy sets)) ε-covers the universe of discourse, if for all the elements in the

universe of discourse a linguistic term exists, which have a membership value grater or equal to ε.

Complete (or Dense) Fuzzy Rule Base: A fuzzy rule base is complete, or dense if all the input universes are ε-covered by rule antecedents, where ε>0. In case of Complete Fuzzy Rule Base, for all the possible multidimensional observations, a rule antecedent must exist, which has a nonzero activation degree. Note, that completeness of the fuzzy rule base is not equivalent with covering fuzzy partitions on each antecedent universe (required but not sufficient in multidimensional case). Usually the number of the rules of a complete rule base is $O(M^I)$, where M is the average number of the linguistic terms in the fuzzy partitions and I is the number of the input universe.

Convex and Normal Fuzzy (CNF) Set: A fuzzy set defined on a universe of discourse holds total ordering, which has a height (maximal membership value) equal to one (i.e. normal fuzzy set), and having membership grade of any elements between two arbitrary elements grater than, or equal to the smaller membership grade of the two arbitrary boundary elements (i.e. convex fuzzy set).

Fuzzy Compositional Rule of Inference (CRI): The most common fuzzy inference method. The fuzzy conclusion is calculated as the fuzzy composition (Klir & Folger, 1988) of the fuzzy observation and the fuzzy rule base relation (see "Fuzzy dot" representation of fuzzy rules). In case of the Zadeh - Mamdani - Larsen max-min compositional rule of inference (Zadeh, 1973) (Mamdani, 1975) (Larsen, 1980) the applied fuzzy composition is the max-min composition of fuzzy relations ("max" stands for the applied s-norm and "min" for the applied t-norm fuzzy operations).

"Fuzzy Dot" Representation of Fuzzy Rules: The most common understanding of the If-Then fuzzy rules.

The fuzzy rules are represented as a fuzzy relation of the rule antecedent and the rule consequent linguistic terms. In case of the Zadeh - Mamdani - Larsen compositional rule of inference (Zadeh, 1973) (Mamdani, 1975) (Larsen, 1980) the fuzzy rule relations are calculated as the fuzzy cylindric closures (t-norm of the cylindric extensions) (Klir & Folger, 1988) of the antecedent and the rule consequent linguistic terms.

Fuzzy Rule Interpolation: A way for fuzzy inference by interpolation of the existing fuzzy rules based on various distance and similarity measures of fuzzy sets. A suitable method for handling sparse fuzzy rule bases, since FRI methods can provide reasonable (interpolated/extrapolated) conclusions even if none of the existing rules fires under the current observation.

Sparse Fuzzy Rule Base: A fuzzy rule base is sparse, if an observation may exist, which hits no rule antecedent. (The rule base is not complete.)

Vague Environment (VE): The idea of a VE is based on the similarity (or in this case the indistinguishability) of the considered elements. In VE the fuzzy membership function $\mu_A(x)$ is indicating level of similarity of x to a specific element a that is a representative or prototypical element of the fuzzy set $\mu_A(x)$, or, equivalently, as the degree to which $x \in X$ is indistinguishable from $a \in X$ (Klawonn, 1994). Therefore the α-cuts of the fuzzy set $\mu_A(x)$ are the sets which contain the elements that are $1-\alpha$ -indistinguishable from a. Two values in a VE are ε-distinguishable if their distance is greater than ε. The distances in a VE are weighted distances. The weighting factor or function is called *scaling function (factor)* (Klawonn, 1994). If the VE of a fuzzy partition (the scaling function or at least the approximate scaling function (Kovács, 1996), (Kovács & Kóczy, 1997b)) exists, the member sets of the fuzzy partition can be characterized by points in that VE.

Fuzzy Systems Modeling: An Introduction

Young Hoon Joo
Kunsan National University, Korea

Guanrong Chen
City University of Hong Kong, China

INTRODUCTION

The basic objective of system modeling is to establish an input-output representative mapping that can satisfactorily describe the system behaviors, by using the available input-output data based upon physical or empirical knowledge about the structure of the unknown system.

BACKGROUND

Conventional system modeling techniques suggest constructing a model described by a set of differential or difference equations. This approach is effective only when the underlying system is mathematically well-defined and precisely expressible. They often fail to handle uncertain, vague or ill-defined physical systems, and yet most real-world problems do not obey such precise, idealized, and subjective mathematical rules. According to the incompatibility principle (Zadeh, 1973), as the complexity of a system increases, human's ability to make precise and significant statements about its behaviors decreases, until a threshold is reached beyond which precision and significance become impossible. Under this principle, Zadeh (1973) proposed a modeling method of human thinking with fuzzy numbers rather than crisp numbers, which had eventually led to the development of various fuzzy modeling techniques later on.

MAIN FOCUS OF THE CHAPTER

Structure Identification

In structure identification of a fuzzy model, the first step is to select some appropriate input variables from the collection of possible system inputs; the second step is to determine the number of membership functions for each input variable. This process is closely related to the partitioning of input space. Input space partitioning methods are useful for determining such structures (Wang & Mendel, 1996).

Grid Partitioning

Figure 1 (a) shows a typical grid partition in a two-dimensional input space. Fuzzy grids can be used to generate fuzzy rules based on system input-output training data. Also, a one-pass build-up procedure can avoid the time-consuming learning process, but its performance depends heavily on the definition of the grid. In general, the finer the grid is, the better the performance will be. Adaptive fuzzy grid partitioning can be used to refine and even optimize this process. In the adaptive approach, a uniformly partitioned grid may be used for initialization. As the process goes on, the parameters in the antecedent membership functions will be adjusted. Consequently, the fuzzy grid evolves. The gradient descent method may then be used to optimize the size and location of the fuzzy grid regions and the overlapping degree among them. The major drawback of this grid partition method is that the performance suffers from an exponential explosion of the number of inputs or membership functions as the input variables increase, known as the "curse of dimensionality," which is a common issue for most partitioning methods.

Tree Partitioning

Figure 1 (b) visualizes a tree partition. The tree partitioning results from a series of guillotine cuts. Each region is generated by a guillotine cut, which is made entirely across the subspace to be partitioned. At the $(k-1)$st iteration step, the input space is partitioned into k regions. Then a guillotine cut is applied to one of

these regions to further partition the entire space into $k + 1$ regions. There are several strategies for determining which dimension to cut, where to cut at each step, and when to stop. This flexible tree partitioning algorithm resolves the problem of curse of dimensionality. However, more membership functions are needed for each input variable, and they usually do not have clear linguistic meanings; moreover, the resulting fuzzy model consequently is less descriptive.

Scatter Partitioning

Figure 1 (c) illustrates a scatter partition. This method extracts fuzzy rules directly from numerical data (Abe & Lan, 1995). Suppose that a one-dimensional output, y, and an m-dimensional input vector, $\underline{x}$, are available. First, the output space is divided into n intervals, $[y_0, y_1]$, $(y_1, y_2]$, …, $(y_{n-1}, y_n]$, where the ith interval is called "output interval i." Then, activation hyperboxes are determined, which define the input region corresponding to the output interval i, by calculating the minimum and maximum values of the input data for each output interval. If the activation hyperbox for the output interval i overlaps with the activation hyperbox for the output interval j, then the overlapped region is defined as an inhibition hyperbox. If the input data for output intervals i and/or j exist in the inhibition hyperbox, then within this inhibition hyperbox one or two additional activation hyperboxes will be defined. Moreover, if two activation hyperboxes are defined and they overlap, then an additional inhibition hyperbox

is further defined. This procedure is repeated until overlapping is resolved.

Parameters Identification

After the system structure has been determined, parameters identification is in order. In this process, the optimal parameters of a fuzzy model that can best describe the input-output behavior of the underlying system are searched by optimization techniques.

Sometimes, structure and parameters are identified under the same framework through fuzzy modeling. There are virtually many different approaches to modeling a system using the fuzzy set and fuzzy system theories (Chen & Pham, 1999, 2006), but the classical least-squares optimization and the general Genetic Algorithm (GA) optimization techniques are most popular. They are quite generic, effective, and competitive with other successful non-fuzzy types of optimization-based modeling methods such as neural networks and statistical Monte Carlo.

An Approach Using Least-Squares Optimization

A fuzzy system can be described by the following generic form:

$$f(\underline{x}) = \sum_{k=1}^{m} \alpha_k g_k(\underline{x}) = \underline{\alpha}^T \underline{g}(\underline{x})$$

(1)

Figure 1. Three typical MISO partitioning methods

| *(a) fuzzy grid* | *(b) tree partition* | *(c) scatter partition* |
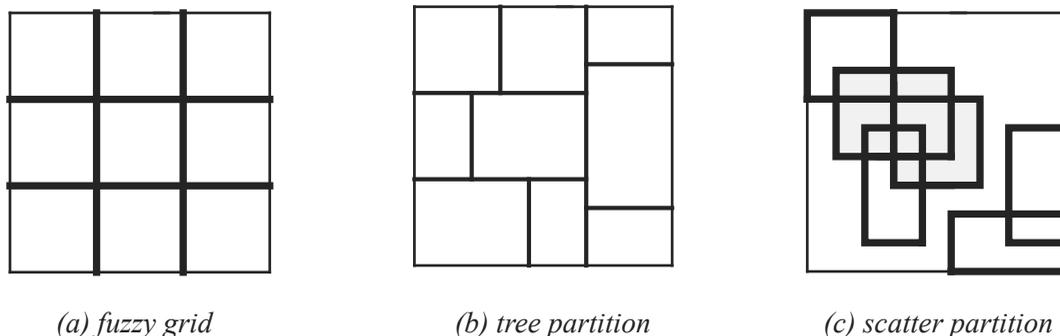
where $\underline{\alpha} = [\alpha_1, \cdots, \alpha_m]^T$ are constant coefficients and

$$g_k(\underline{x}) = \frac{\Pi_{i=1}^n \mu_{X_{kj}}(\underline{x})}{\sum_{k=1}^m \left(\Pi_{i=1}^n \mu_{X_{kj}}(\underline{x})\right)}, \; k = 1, \cdots, m \qquad (2)$$

are the basis functions, in which $\mu_{X_{kj}}(\cdot)$ are the chosen membership functions. Suppose that the real system output is

$$y(t) = \sum_{k=1}^m \alpha_k g_k(\underline{x}) + e(t) \qquad (3)$$

where $y(t)$ is the system output and $e(t)$ represents the modeling error, which is assumed to be uncorrelated with the fuzzy basis functions $\{g_k(\cdot)\}_{k=1}^m$ in this discussion.

Suppose that n pairs of system input-output data are given: $(\underline{x}_d(t_i), y_d(t_i))$, $i = 1, \cdots, n$. The goal is to find the best possible fuzzy basis functions, such that the total least-squares error between the data set and the system outputs $\{y(t_i)\}_{i=1}^n$ is minimized. To do so, the linear model (3) is first written in a matrix form over the time domain $t_1 < \cdots < t_n$, namely,

$$\underline{y} = G\underline{\alpha} + \underline{e}$$

where $\underline{y} = [y(t_1), \cdots, y(t_n)]^T$, $\underline{e} = [e(t_1), \cdots, e(t_n)]^T$, and

$$G := [\underline{g}_1, \cdots, \underline{g}_n]^T = \begin{bmatrix} g_1(t_1) & \cdots & g_m(t_1) \\ \vdots & & \vdots \\ g_1(t_n) & \cdots & g_m(t_1) \end{bmatrix}$$

with $\underline{g}_j = [g_j(t_1), \cdots, g_j(t_n)]^T$, $j = 1, \cdots, n$.

The first step is to transform the set of numbers, $g_i(t_j)$, $i = 1, \cdots, m$, $j = 1, \cdots, n$, into a set of orthogonal basis vectors, and only significant basis vectors are used to form the final least-squares optimization. Here, the Gaussian membership functions

$$\mu_{X_{kj}}(x_k) = c_{kj} \exp\left\{-\left(x_k - \overline{x}_{kj}/\sigma_{kj}\right)^2/2\right\}$$

are used as an example to illustrate the computational algorithm.

One approach to initializing the fuzzy basis functions is to choose n initial basis functions, $g_k(x)$, in the form of (2) with $m = n$ in this discussion, and initially with $c_{kj} = 1$, $\overline{x}_{kj} = x_k(t_j)$, and

$$\sigma_{kj} = \frac{1}{m_l}\Big[\max\{x_k(t_j), j = 1, \cdots, n\} -$$

$$\min\{x_k(t_j), j = 1, \cdots, n\}\Big], k = 1, \cdots, n$$

where m_l is the number of the basis functions in the final expression, which is determined by the designer based on experience (usually, $m_l < n$).

After choosing the initial fuzzy basis functions, the next step is to select the most significant ones among them. This process is based on the classical Gram-Schmidt orthogonalization, while c_{kj}, $\overline{x}_{kj}$, and σ_{kj} are all fixed:

Step 1. For $j = 1$, compute

$$\underline{w}_1^{(i)} = g_i(\underline{x}_d) = [g_i(x_d(t_1)), \cdots, g_i(x_d(t_n))]^T$$

$$h_1^{(i)} = \frac{(\underline{w}_1^{(i)})^T \underline{y}_d}{(\underline{w}_1^{(i)})^T \underline{w}_1^{(i)}}$$

$$\varepsilon_{1i} = (h_1^{(i)})^2 \frac{(\underline{w}_1^{(i)})^T \underline{w}_1^{(i)}}{\underline{y}_d^T \underline{y}_d}$$

$(1 \leq i \leq n)$ where

$$\underline{x}_d = [x_d(t_1), \cdots, x_d(t_n)]^T \text{ and}$$

$$\underline{y}_d = [y_d(t_1), \cdots, y_d(t_n)]^T$$

are the input-output data set. Then, compute

$$\varepsilon_1^{(i_1)} = \max\{\varepsilon_1^{(i)} : 1 \leq i \leq n\}$$

and let

$$\underline{w}_1 = \underline{w}_1^{(i_1)} = \underline{g}_{i_1} \text{ and } h_1 = h_1^{(i_1)}.$$

Step 2. For each j, $2 \leq j \leq m_l$, compute

$$c_{kj}^{(i)} = \frac{\underline{w}_k^T \underline{g}_i}{\underline{w}_k^T \underline{w}_k}$$

$$\underline{w}_j^{(i)} = \underline{g}_i - \sum_{k=1}^{j-1} c_{kj}^{(i)} \underline{w}_k$$

$$h_j^{(i)} = \frac{(\underline{w}_j^{(i)})^T \underline{y}_d}{(\underline{w}_j^{(i)})^T \underline{w}_j^{(i)}}$$

$$\varepsilon_j^{(i)} = (h_j^{(i)})^2 \frac{(\underline{w}_j^{(i)})^T \underline{w}_j^{(i)}}{\underline{y}_d^T \underline{y}_d}$$

$$\varepsilon_k^{(i_j)} = \max \left\{ \varepsilon_j^{(i)} : \ 1 \le i \le n; \ i \ne i_1, \cdots, i \ne i_{j-1} \right\}$$

where $\varepsilon_j^{(i)}$ represents the error-reduction ratio due to $\underline{w}_j^{(i)}$. Pick

$$\underline{w}_j = \underline{w}_j^{(i_j)} \text{ and } h_k = h_k^{(i_j)}.$$

Step 3. Solve equation

$$A^{(m_l)} \underline{\alpha}^{(m_l)} = \underline{h}^{(m_l)}$$

for a solution $\underline{\alpha}^{(m_l)} = [\alpha_1^{(m_l)}, \cdots, \alpha_{m_l}^{(m_l)}]^T$,

where $\underline{h}^{(m_l)} = [h_1, \cdots, h_{m_l}]^T$ and

$$A^{(m_l)} = \begin{bmatrix} 1 & c_{12}^{(i_2)} & c_{13}^{(i_3)} & \cdots & c_{1m_l}^{(i_{m_l})} \\ 0 & 1 & c_{23}^{(i_3)} & \cdots & c_{2m_l}^{(i_{m_l})} \\ \vdots & 0 & \ddots & & \vdots \\ 0 & & 0 & 1 & c_{m_l-1,m_l}^{(i_{m_l})} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

The final result is obtained as

$$f(\underline{x}) = \sum_{k=1}^{m_l} \alpha_k^{(m_l)} g_{i_k}(\underline{x})$$

An Approach Using Genetic Algorithms

The parameter identification procedure is generally very tedious for a large-scale complex system, for which the GA approach has some attractive features such as its great flexibility and robust optimization ability (Man, Tang, Kwong & Halang, 1997).

GA is attributed to Holland (1975), which was applied to fuzzy modeling and fuzzy control in the 1980s.

GA can be used to find an optimal or suboptimal fuzzy model to describe a given system without manual design (Joo, Hwang, Kim & Woo, 1997; Liska & Melsheimer, 1994; Soucek & Group, 1992). In addition, GA fuzzy modeling method can be integrated with other components of a fuzzy system, so as to achieve overall superior performance in control and automation.

Genetic Algorithm Preliminaries

GA provides an optimization method, with a stochastic search algorithm, based on some common biological principles of selection, crossover and mutation. A GA algorithm encodes each point in a solution space into a string composing of binary or real values, called a *chromosome*. Each point is assigned a fitness value from zero to one, which is usually taken to be the same as the objective function to be maximized. A GA scheme keeps a set of points as a *population*, which is evolved repeatedly toward a better and possibly the best fitness value. In each *generation*, GA generates a new population using genetic operators such as crossover and mutation. Through these operations, individuals with higher fitness values are more likely to survive and to participate in the next genetic operations. After a number of generations, individuals with higher fitness values are kept in the population while the others are eliminated. GA, therefore, can ensure a gradual increasing of improving solutions, till a desired optimal or suboptimal solution is obtained.

Basic GA Elements

A simple genetic algorithm (SGA) was first described by Goldberg (1989) and is used here for illustration, with a pseudo-code shown below, where the population at time t is a time function, $P = P(t)$, with a random initial population $P(0)$.

```
Procedure GA
Begin
t = 0
Initialize P(t)
Evaluate P(t)
While not finished do
     Begin
      t = t + 1
      Reproduce P(t) from P(t − 1)
      Crossover individuals in P(t)
```

    Mutate individuals in $P(t)$
    Evaluate $P(t)$
    End
End

Population Representation and Initialization

Individuals are encoded as *strings* (i.e., chromosomes) composing of some alphabets, so that the *genotypes* (chromosome values) are uniquely mapped onto the decision variable (*phenotype*) domain. The most commonly used representation in GA is the binary alphabet, {0,1}; others are ternary, integer, real-valued, etc. (Takagi & Sugeno, 1985).

The search process, described below, will operate on these encoding decision variables rather than the decision variables themselves, except when real-valued genes are used. After a representation method has been chosen to use, the first step in the SGA is to create an initial population, by generating the required number of individuals via a random number generator which uniformly distributes initial numbers in the desired range.

Objective and Fitness Functions

The *objective function* is used to measure the performance of the individuals over the problem domain. The *fitness function* is used to transform the objective function value into a measure of relative fitness; mathematically, $F(x) = g(f(x))$, where f is the objective function, g is the transform that maps the value of f to a nonnegative number, and F is the resulting relative fitness. In general, the fitness function value corresponds to the number of offspring, and an individual can expect to produce this value in the next generation. A commonly used transform is the proportional fitness assignment, defined by

$$F(x_i) = f(x_i)/\sum_{i=1}^{N} f(x_i),$$

where N is the population size and x_i is the phenotypic value of individual i, $i = 1,\cdots, N$.

Although the above fitness assignment ensures that each individual has a certain probability of reproduction according to its relative fitness, it does not account for negative objective function values. A linear transform, which offsets the objective function, is often used prior to the fitness assignment. It takes the form

$$F(x) = fa(x) + b,$$

where a is a positive scaling factor if the optimization is to maximize the objective function but is negative if it is a minimization, and the offset b is used to ensure that the resulting fitness values are all negative.

Then, the selection algorithm selects individuals for reproduction on the basis of their relative fitness.

Reproduction

Once each individual has been assigned a fitness value, they can be chosen from the population with a probability according to their relative fitness. They can then be recombined to produce the next generation.

Most widely used genetic operators in GA are selection, crossover, and mutation operators. They are often run simultaneously in an GA program.

Selection

Selection is the process of determining the number of trials in which a particular individual is chosen for reproduction. Thus, it is the number of offspring that an individual will produce in the *mating pool*, a temporary population where crossover and mutation operations are applied to each individual. The selection of individuals has two separate processes:

a.   determination of the number of trials an individual can expect to receive;
b.   conversion of the expected number of trials into a discrete number of offspring.

Crossover (Recombination)

The crossover operator defines the procedure for generating children from two parents. Analogous to biological crossover, it exchanges genes at a randomly selected crossover point from also randomly selected parents in the mating pool to generate children.

A common method is the following: Parent chromosomes are cut at randomly selected points, which can be more than one, to exchange their genes at some specified crossover points with a user-specified crossover probability. This crossover method is categorized into single-point crossover and multi-point crossover

according to the number of crossover points. Uniform crossover often works well with small populations of chromosomes and for simpler problems (Soucek & Group, 1992).

Mutation

Mutation operation is randomly applied to individuals, so as to change their gene value with a mutation probability, P_m, which is very low in general.

GA Parameters

The choice of the mutation probability P_m and the crossover probability P_c as two control parameters can be a complex nonlinear optimization problem. Their settings are critically dependent upon the nature of the objective function. This selection issue still remains open to better resolutions. One suggestion is that for large population size (say 100), crossover rate is 0.6 and mutation rate is 0.001, while for small population size (such as 30), crossover rate is 0.9 and mutation rate is 0.01 (Zalzala & Fleming, 1997).

GA-Based Fuzzy System Modeling

In GA, parameters for a given problem are represented by the chromosome. This chromosome may contain one or more substrings. Each chromosome contains a possible solution to the problem. Fitness function is used to evaluate how well a chromosome solves the problem. In the GA-based approach for fuzzy modeling, each chromosome represents a specific fuzzy model, and the ultimate goal is to carefully design a good (ideally optimal) chromosome to represent a desired fuzzy model.

Chromosome Structure

As an example, consider a simple fuzzy model with only one rule, along with the scatter partition to be encoded to a chromosome.

Suppose that both real number coding and integer number coding are used. The structure and the parameters of the fuzzy model are encoded into one or more substrings in the chromosome. A chromosome is composed of two substrings (candidate substring and decision substring) and these substrings are divided

Figure 2. A chromosome structure for fuzzy modeling
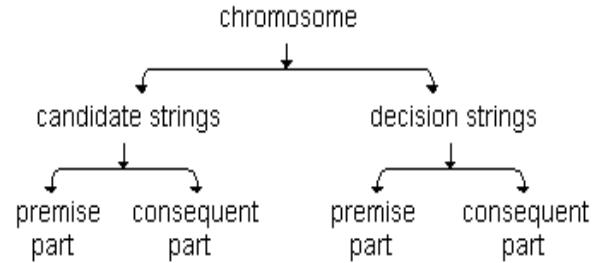
into two parts (IF part and THEN part), as shown in Fig. 2.

The candidate substring is encoded by real numbers, as shown in Fig. 3 (a). It contains the candidates for the parameters of a membership function in the IF part, and the fuzzy singleton membership function in the THEN part. Figure 3 describes the coding format of a candidate substring in a chromosome, where n is the number of input variables, r the number of candidates for parameters in the IF part, and s the number of candidates for the real numbers in the THEN part.

Decision substrings are encoded by integers, which determine the structure and the number of rules, by choosing one of the parameters in the candidate substrings, as illustrated by Fig. 3 (b).

The decision substrings for the IF part determine the premise structure of the fuzzy rule base. It is composed of n genes that take integer values (alleles) between 0 and r. According to this value, an appropriate parameter in the candidate substring is selected. A zero value means that the related input is not included in the rule. A decision substring for the THEN part is composed of c (the maximum number of rules) genes that take the integer values between 0 and s, which chooses appropriate values from the candidate substring for the THEN part. In this substring, the gene taking the zero value deletes the related rule. Therefore, these substrings determine the structure of the THEN part and the number of rules. Figure 4 illustrates an example of decoding the chromosome, with the resulting fuzzy rule shown in Fig. 5.

Figure 3. Two basic functions in a chromosome
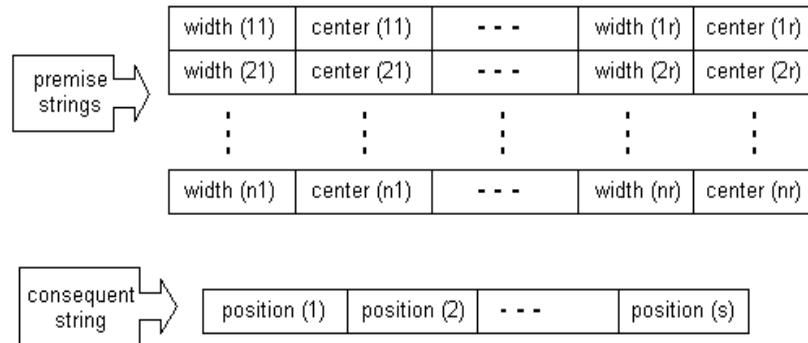
(a) The candidate substrings
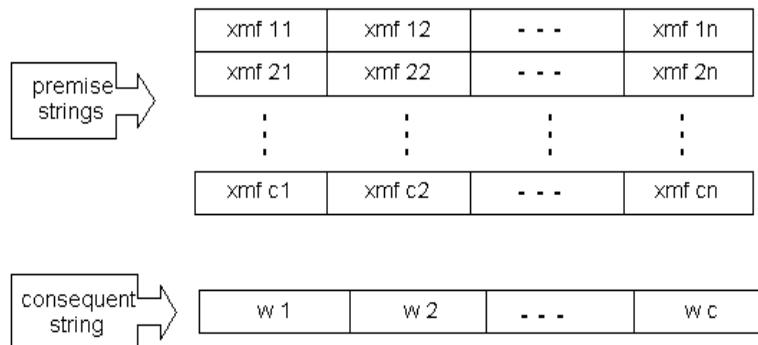
(b) The decision substrings

Fitness Function

To measure the performance of the GA-based fuzzy modeling, an objective function is defined for optimization, which is chosen by the designer and usually is a least-squares matching measure of the form

$$J = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^d)^2 ,$$

where $\{y_i\}$ and $\{y_i^d\}$ are the fuzzy model outputs and desired outputs, respectively, and n is the number of the data used.

Since GA is guided by the fitness values and requires literally no limit on the formulation of its performance measure, one can incorporate more information about a fuzzy model into the fitness function: $f = g(J_{structure}, J_{accuracy}, \cdots)$. One example of a fitness function is

$$f(J) = \frac{\lambda}{J} + \frac{1-\lambda}{1+c} ,$$

where $\lambda \in [0,1]$ is the weighting factor (a large λ gives a highly accurate model but requires a large number of rules), and c is the maximum number of rules. When the fitness function is evaluated over an empty set, it is

F

Figure 4. An example of genetic decoding process
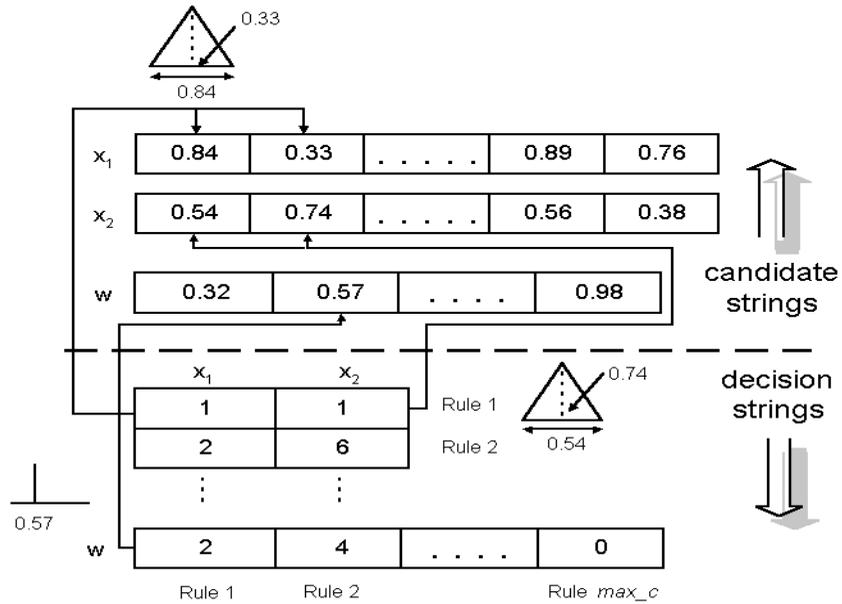
Figure 5. The first fuzzy rule obtained by the decoding processes
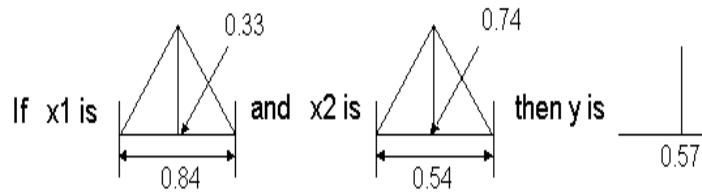
undefined; but in this case one may introduce a penalty factor, $0 < p < 1$, and compute $p \cdot f(J)$ instead of $f(J)$. If an individual with a very high fitness value appears at the earlier stage, this fitness function may cause early convergence of the solution, thereby stopping the algorithm before optimality is reached. To avoid this situation, the individuals may be sorted according to their raw fitness values, and the new fitness values are determined recursively by

$$f_1 = 1, f_2 = fa_1 = a, ..., f_m = a^m$$

for a fitness scaling factor $a \in (0,1)$.

GA-Based Fuzzy Modeling with Fine Tuning

GA generally does not guarantee the convergence to a global optimum. In order to improve this, the gradient descent method can be used to fine tune the parameters

identified by GA. Since GA usually can find a near global optimum, to this end fine tuning of the membership function parameters in both IF and THEN parts, e.g., by a gradient descent method, can generally lead to a global optimization (Chang, Joo, Park & Chen, 2002; Goldberg, 1989).

FUTURE TRENDS

This will be further discussed elsewhere in the future.

CONCLUSION

Fuzzy systems identification is an important and yet challenging subject for research, which calls for more efforts from the control theory and intelligent systems communities, to reach another high level of efficiency and success.

REFERENCES

S. Abe & M. S. Lan (1995). Fuzzy rules extraction directly from numerical data for function approximation. *IEEE Trans. on Systems, Man and Cybernetics*. 25: 119-129.

W. Chang, Y. H. Joo, J. B. Park & G. Chen (2002). Design of robust fuzzy-model-based controller with sliding mode control for SISO nonlinear systems. *Fuzzy Sets and Systems*. 125:1-22.

G. Chen & T. T. Pham (1999). *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. CRC Press.

G. Chen & T. T. Pham (2006). *Introduction to Fuzzy Systems*. CRC Press.

E. Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

J. H. Holland (1975). *Adaptation in Natural and Artificial Systems*. MIT Press.

Y. H. Joo, H. S. Hwang, K.B. Kim & K.B. Woo (1997) Fuzzy system modeling by fuzzy partition and GA hybrid schemes. *Fuzzy Sets and Systems*. 86: 279-288.

J. Liska & S. S. Melsheimer (1994). Complete design of fuzzy logic systems using genetic algorithms. *Proc. of IEEE Conf. on Fuzzy Systems*. 1377-1382.

K. F. Man, K. S. Tang, S. Kwong & W. A. Halang (1997). *Genetic Algorithms for Control and Signal Processing*. Springer.

B. Soucek & T. I. Group (1992). *Dynamic Genetic and Chaotic Programming*. Wiley.

W. Spears & V. Anand (1990). The use of crossover in genetic programming. *NRL Technical Report*, AI Center, Naval Research Labs, Washington D. C.

T. Takagi & M. Sugeno (1985) Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man and Cybernetics*. 15: 116-132.

L. X. Wang & J. M. Mendel (1996). Generating fuzzy rules by learning from examples. *IEEE Trans. on Systems, Man and Cybernetics*. 22:1414-1427.

L. A. Zadeh (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. on Systems, Man and Cybernetics*. 3: 28-44.

A. M. S. Zalzala & P. J. Fleming (1997) *Genetic Algorithms in Engineering Systems*. IEE Press.

KEY TERMS

Fuzzy Rule: A logical rule established based on fuzzy logic.

Fuzzy System: A system formulated and described by fuzzy set-based real-valued functions.

Genetic Algorithm: An optimization scheme based on biological genetic evolutionary principles.

Least-Squares Algorithm: An optimization scheme that minimizes the square of the sum of the approximation errors.

Parameter Identification: Find appropriate parameter values in a mathematical model.

Structure Identification: Find a mathematical representation of the unknown system's structure.

System Modeling: A mathematical formulation of an unknown physical system or process.

F

Gene Regulation Network Use for Information Processing

Enrique Fernandez-Blanco
University of A Coruña, Spain

J. Andrés Serantes
University of A Coruña, Spain

INTRODUCTION

From the unicellular to the more complex pluricellular organism needs to process the signals from its environment to survive. The computation science has already observed, that fact could be demonstrated remembering the artificial neural networks (ANN). This computation tool is based on the nervous system of the animals, but not only the nervous cells process information in an organism. Every cell has to process the development and functioning plan encoded at its DNA and every one of these cells executes this program in parallel with the others. Another interesting characteristic of natural cells is that they form systems that are tolerant to partial failures: small errors do not induce a global collapse of the system.

The present work proposes a model that is based on DNA information processing, but adapting it to general information processing. This model can be based on a set of techniques called Artificial Embryogeny (Stanley K. & Miikkulainen R. 2003) which adapts characteristics from the biological cells to solve different problems.

BACKGROUND

The Evolutionary Computation (EC) field has given rise to a set of models that are grouped under the name of Artificial Embryology (AE), first introduced by Stanley and Miikkulainnen (Stanley K. & Miikkulainen R. 2003). This group refers to all the models that try to apply certain characteristics of biological embryonic cells to computer problem solving, i.c. self-organisation, failure tolerance, and parallel information processing.

The work on AE has two points of view. On the one hand can be found the grammatical models based on L-systems (Lindenmayer A. 1968) which do a top-down

approach to the problem. On the other hand can be found the chemical models based on the Turing's ideas (Turing A. 1952) which do a down-top approach.

The grammatically approach, some times, has used the models for study the evolution of ANN, which is known as neuroevolution. The first neuroevolution system was development by Kitano (Kitano, H. 1990). In this work Kitano shows that it was possible to evolve the connectivity matrix of ANN through a set of rewrite rules. Another remarkable work is the application of L-systems do by Hornby and Pollack (Hornby, G. S. & Pollack J. B. 2002). At this work they simultaneously evolved the body morphologies and the neural network of artificial creatures in a simulated 3D physical environment. Finally, mention the works carry out by Gruau (Gruau F. 1994) where the author uses grammar trees to encode steps in the development of a neural network from a single antecesor cell.

On the chemical approach, the starting point of this field can be found in the modelling of gene regulatory networks, performed by Kauffmann in 1969 (Kauffman S.A. 1969). After that, several works were carried out on subjects such as the complex behaviour generated by the fact that the differential expression of certain genes has a cascade influence on the expressions of others (Mjolsness E., Sharp D.H., & Reinitz J. 1995). Considering the gene regulatory networks works, the most relevant models are the following: the Kumar and Bentley model (Kumar S. & Bentley P.J 2003), which uses the theory of fractal proteins Bentley, P.J., Kumar, S. 1999; for the calculation of protein concentration; the Eggenberger model (Eggenberger P. 1996), which uses the concepts of cellular differentiation and cellular movement to determine cell connections; and the work of Dellaert and Beer (Dellaert F. & Beer R.D. 1996), who propose a model that incorporates the idea of biological operons to control the model expression, where the function assumes the mathematical meaning of a Boolean function.

GENETIC REGULATORY NETWORK MODEL

The cells of a biological system are mainly determined by the DNA strand, the genes, and the proteins contained by the cytoplasm. The DNA is the structure that holds the gene-encoded information that is needed for the development of the system. The genes are activated or transcribed thanks to the protein shaped-information that exists in the cytoplasm, and consist of two main parts: the sequence, which identifies the protein that will be generated if the gene is transcribed, and the promoter, which identifies the proteins that are needed for gene transcription.

Another remarkable aspect of biological genes is the difference between constitutive genes and regulating genes. The latter are transcribed only when the proteins identified in the promoter part are present. The constitutive genes are always transcribed, unless inhibited by the presence of the proteins identified in the promoter part, acting then as gene oppressors.

The present work has tried to partially model this structure with the aim of fitting some of its abilities into a computational model; in this way, the system would have a structure similar that is similar to the above and will be detailed in the next section.

Proposed Model

Various model variants were developed on the basis of biological concepts. The proposed artificial cellular system is based on the interaction of artificial cells by means of messages that are called proteins. These cells can divide themselves, die, or generate proteins that will act as messages for themselves as well as for neighbour cells.

The system is supposed to express a global behaviour towards the information processing. Such behaviour would emerge from the information encoded in a set of variables of the cell that, in analogy with the biological cells, will be named genes.

The central element of our model is the artificial cell. Every cell has a binary string-encoded information for the regulation of its functioning. Following the biological analogy, this string will be called DNA. The cell also has a structure for the storage and management of the proteins generated by the own cell and those received from neighbourhood cells; following the biological model, this structure is called cytoplasm.

The DNA of the artificial cell consists of functional units that are called genes. Each gene encodes a protein or message (produced by the gene). The structure of a gene has four parts (see **Figure 1**):

- Sequence: the binary string that corresponds to the protein that encodes the gene
- Promoters: is the gene area that indicates the proteins that are needed for the gene's transcription.
- Constituent: this bit identifies if the gene is constituent or regulating
- Activation percentage (binary value): the percentage of minimal concentration of promoters proteins inside the cell that causes the transcription of the gene.

The transcription of the encoded protein occurs when the promoters of the non-constituent genes appear in a certain rate at the cellular cytoplasm. On the other hand, the constituent genes are expressed until such expression is inhibited by the present rate of the promoter genes.

The other fundamental element for keeping and managing the proteins that are received or produced by the artificial cell is the cytoplasm. The stored proteins have a certain life time before they are erased. The cytoplasm checks which and how many proteins are needed for the cell to activate the DNA genes, and as such responds to all the cellular requirements for the concentration of a given type of protein. The cytoplasm also extracts the proteins from the structure in case they are needed for a gene transcription.

Figure 1. Structure of a system gene

Figure 2. Logical operators match
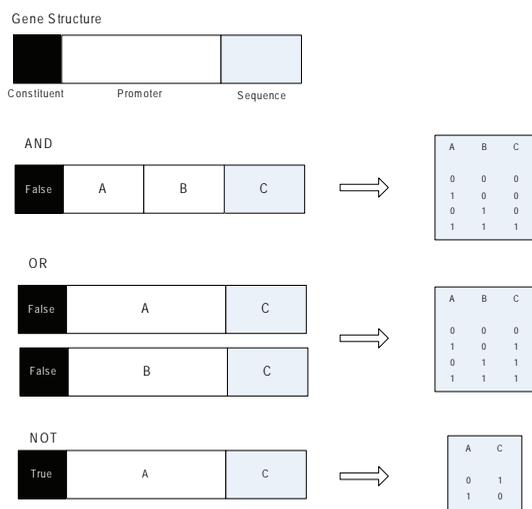
The Information Processing Capacities

The biological cells, besides generating structures, work as small processors for parallel information handling with the remaining cells. The information that they process comes from their own generation and from their environment. On the basis of this fact, the present work has explored the generation capabilities of the model structure, although using the gene and protein structure, an operation set with Boolean algebra-like structure might be defined.

The space for the definition of the operations would be the presence or absence of certain proteins into the system, whereas the operation result would be the protein contained/encoded at the gene. The AND operation (see Figure 2) would be modelled with a gene that would need for its expression all the proteins of its promoters. The OR operation would be modelled with two genes that, despite their different promoters, result in the same protein. Finally, the NOT operation would be modelled with the constituent part, which changes the performance of that gene. The presence of proteins belonging to the promoters would imply the absence of the gene resulting protein at the system. This behaviour is similar to the gene regulatory networks (Kauffman S.A. 1969).

The Artificial Neuron Networks (ANNs) can be configured for carrying out these processing tasks.

This analogous functioning seems to indicate that the system could execute more complex tasks, as ANNs do (Hassoun M.H. 1995).

FUTURE TRENDS

The final objective of this group is to develop an artificial model which is based on the biologically model with a processing information capacity similar to the ANN. In order to archive this objective some simple tests have been developed to check the functioning of the model. The result of these tests show that is possible to process information using the gene regulatory network as the basing system.

From this point of development, the next steps of development must go in order to develop more complex task and to study the functioning of the model. Other objective for future works can be the combination of the process information capacities of the model with the generating structure capacities presented in (Fernández-Blanco E., Dorado J., Rabuñal J.R., Gestal M. & Pedreira N. 2007).

CONCLUSION

At this work some properties of biological cells have been adapted to an artificial model. In particular the gene regulatory network idea was adapted to processing information. This adaptation has its bases on using the transcription rule to determine a boolean algebra like structure. The result of this adaptation is that, now, we can use it to develop processing information tests and.

Finally comment that this new way of generation processing information networks has a lot of test and studies to do until it is stabilized as a consolidated technique for information processing.

REFERENCES

Bentley, P.J., Kumar, S. (1999) *Three ways to grow designs: A comparation of three embryogenies for an evolutionary design problem.* In Proceedings of Genetic and Evolutionay Computation.

G

Dellaert F. & Beer R.D. (1996) *A Developmental Model for the Evolution of Complete Autonomous Agent* In From animals to animats: Proceedings of the Forth International Conference on Simulation of Adaptive Behavior, Massachusetts, September 9-13, pp. 394-401, MIT Press.

Eggenberger P. (1996) *Cell Interactions as a Control Tool of Developmental Processes for Evolutionary Robotics*. In From animals to animats: Proceedings of the Forth International Conference on Simulation of Adaptive Behavior, Massachusetts, September 9-13, pp. 440-448, MIT Press.

Fernández-Blanco E., Dorado J., Rabuñal J.R., Gestal M. & Pedreira N. (2007) *A New Evolutionary Computation Technique for 2D Morphogenesis and Information Processing*. WSEAS Transactions on Information Science & Applications vol. 4(3) pp.600-607, WSEAS Press.

Gruau F. (1994)*Neural network synthesis using cellular encoding and the genetic algorithm*. Doctoral dissertation, Ecole Normale Superieure de Lyon, France.

Hassoun M.H. (1995)*Fundamentals of Artificial Neural Networks*. University of Michigan Press, MA, USA

Hornby, G. S. & Pollack J. B. (2002)*Creating high-level components with a generative representation for body brain evolution*. Artificial Life vol.8 issue 3.

Kauffman, S.A. (1969) *Metabolic stability and epigenesis in randomly constructed genetic nets*. Journal of Theoretical Biology 22 pp. 437-467.

Kitano, H. (1990). *Designing neural networks using genetic algorithm with dynamic graph generation system*. Complex Systems vol. 4 pp. 461-476

Kumar, S. & Bentley P.J. (editors) (2003). *On Growth, Form and Computers*. Academic Press. London UK.

Lindenmayer, A. (1968) *Mathematical models for cellular interaction in development: Part I and II*. Journal of Theorical Biology. Vol. 18 pp. 280-299, pp. 300-315.

Mjolsness, E., Sharp, D.H., & Reinitz, J. (1995)*A Connectionist Model of Development*. Journal of Theoretical Biology 176: 291-300.

Stanley, K. & Miikkulainen, R. (2003) *A Taxonomy for Artificial Embryogeny*. In Proceedings Artificial Life 9, pp. 93-130. MIT Press.

Turing, A.(1952) *The chemical basis of morphogenesis*. Philosofical Transactions of the Royal Society B, vol.237, pp. 37-72

KEY TERMS

Artificial Cell: Each of the elements that process the orders codified into the DNA.

Artificial Embryogeny: Under this term are all the processing models which use biological development ideas as inspiration.

Cytoplasm: Part of an artificial cell which is responsible of management the protein-shaped messages.

DNA: Set of rules which are responsible of the cell behaviour.

Gene: Each of the rules which codifies one action of the cell.

Gene Regulatory Network: Term that names the connexion between the different genes of a DNA. The connexion identifies the genes that are necessary for the transcription of other ones.

Protein: This term identifies every kind of the messages that receives an artificial cell.

Genetic Algorithm Applications to Optimization Modeling

Pi-Sheng Deng
California State University at Stanislaus, USA

INTRODUCTION

Genetic algorithms (GAs) are stochastic search techniques based on the concepts of natural population genetics for exploring a huge solution space in identifying optimal or near optimal solutions (Davis, 1991)(Holland, 1992)(Reeves & Rowe, 2003), and are more likely able to avoid the local optima problem than traditional gradient based hill-climbing optimization techniques when solving complex problems.

In essence, GAs are a type of reinforcement learning technique (Grefenstette, 1993), which are able to improve solutions gradually on the basis of the previous solutions. GAs are characterized by their abilities to combine candidate solutions to exploit efficiently a promising area in the solution space while stochastically exploring new search regions with expected improved performance. Many successful applications of this technique are frequently reported across various kinds of industries and businesses, including function optimization (Ballester & Carter, 2004)(Richter & Paxton, 2005), financial risk and portfolio management (Shin & Han, 1999), market trading (Kean, 1995), machine vision and pattern recognition (Vafaie & De Jong, 1998), document retrieval (Gordon, 1988), network topological design (Pierre & Legault, 1998)(Arabas & Kozdrowski, 2001), job shop scheduling (Özdamar, 1999), and optimization for operating system's dynamic memory configuration (Del Rosso, 2006), among others.

In this research we introduce the concept and components of GAs, and then apply the GA technique to the modeling of the batch selection problem of flexible manufacturing systems (FMSs). The model developed in this paper serves as the basis for the experiment in Deng (2007).

GENETIC ALGORITHMS

GAs were simulation techniques proposed by John Holland in the 1960s (Holland, 1992). Basically, GAs solve problems by maintaining and modifying a population of candidate solutions through the application of genetic operators. During this process, beneficial changes to parent solutions are combined into their offspring in developing optimal or near-optimal solutions for the given task.

Intrinsically, GAs explore multiple potentially promising regions in the solution space at the same time, and switch stochastically from one region to another for performance improvement. According to Holland (1992), regions in the solution space can be defined by syntactic patterns of solutions, and each pattern is called a schema. A schema represents the pattern of common attributes or features of the solutions in the same region. Let Σ be an alphabet of symbols. A string over an alphabet is a finite sequence of symbols from the alphabet. An n-ary schema is defined as a string in $(\Sigma \cup \{\#\})^n$, where $\# \notin \Sigma$ is used as a wildcard denotation for any symbol in Σ.

Conceptually, n-ary schemata can be regarded as defining hypersurfaces of an n-dimensional hypercube that represents the space of all n-attribute solutions. Individual solutions in the same region can be regarded as instances of the representing schema, and an individual solution can belong to multiple schemata at the same time. Actually, an n-attribute solution is a member of 2^n different schemata. Therefore, evaluating a solution has the similar effect of sampling 2^n regions (i.e., schemata) at the same time, and this is the famous implicit parallelism of genetic search. A population of M solutions will contain at least 2^n and at most $M \cdot 2^n$ schemata. Even for modest values of n and M, there will be a large number of schemata available for processing in the population. GAs perform an implicit parallel search through the space of possible schemata in the form of performing an explicit parallel search through the space of individual solutions.

The problem solving process of GAs follows a five-phase operational cycle: generation, evaluation, selection, recombination (or crossover), and mutation.

At first a population of candidate solutions is generated. A fitness function or objective function is then defined, and each candidate solution in the population is evaluated to determine its performance or fitness. Based on the relative fitness value, two candidate solutions are selected probabilistically as parents. Recombination is then applied probabilistically to the two parents to form two offspring, and each of the offspring solutions contains some characteristics from its parent solutions. After this, mutation is applied sparingly to components of each offspring solution. The newly generated offspring are then used to replace the low-fitness members in the population. This process is repeated until a new population is formed. Through the above iterative cycles of operations, GAs is able to develop better solutions through progressive generations.

In order to prepare for the investigation of the effects of genetic operations in the sequel of current research, we apply the GA technique to the optimization modeling of manufacturing systems in next section.

A GA-BASED BATCH SELECTION SYSTEM

Batch selection is one of the most critical tasks in the development of a master production plan for flexible manufacturing systems (FMSs). In the manufacturing process, each product requires processing by different sets of tools on different machines with different operations performed in a certain sequence. Each machine has its own limited space capacity in mounting tools and limited amount of available processing time. Under various kinds of resource constraints, choosing an optimal batch of products to be manufactured in a continuous operational process with the purpose to maximize machine utilization or profits has made the batch selection decision a very hard problem. While this problem is usually manageable for manufacturing small number of products, it quickly becomes intractable if the number of products grows even slightly large. The time required to solve the problem exhaustively would grow in a non-deterministic polynomial manner with the number of products to be manufactured.

Batch selection affects all the subsequent decisions in job shop scheduling for satisfying the master production plan, and holds the key to the efficient utilization of resources in generating production plans

for fulfilling production orders. In our formulation, we use the following denotational symbols:

- M: the cardinality of the the set of machines available
- T: the cardinality of the the set of tools available
- P: the cardinality of the set of products to be manufactured
- *MachineUtilization*: the function of total machine utilization
- $processing\_time_{product,tool,machine}$: the time needed to manufacture product *product* using tool *tool* on machine *machine*
- $available\_time_{machine}$: the total available processing time on machine *machine*
- $capacity_{machine}$: the total number of slots available on machine *machine*
- *machine, tool, product*: indicators for machines, tools, and products to be manufactured correspondingly
- $slot_{tool}$: the number of slot required by machine tool *tool*
- $quantity_{product}$: the quantity of product *product* to be manufactured in a shift
- $Q_{product}$: the quantity of product *product* ordered by customers as specified in the production table

Fitness (or Objective) Function

The objective is to identify a batch of products to be manufactured so that the total machine utiliztion rate will be maximized. See Exhibit A.

The above objective function is to be maximized subject to the following resource constraints:

1. Machine capacity constraint (see Exhibit B)

The above function $f(\bullet)$ is used to determine if tool *tool* needs to be mounted on machine *machine* for the processing of the current batch of product.

2. Machine time constraint (see Exhibit C)
3. Non-negativity and integer contraints

Encoder/Decoder

The Encoder/Decoder is a representation scheme used to determine how the problem is structured in the GA

Exhibit A.

$$\text{Maximize} \quad MachineUtilization(quantity_1, quantity_2, ..., quantity_P) =$$

$$\frac{\displaystyle\sum_{machine=1}^{M} \sum_{tool=1}^{T} \sum_{product=1}^{P} processing\_time_{product,tool,machine} quantity_{product}}{\displaystyle\sum_{machine=1}^{M} available\_time_{machine}}$$

Exhibit B.

$$\sum_{tool=1}^{T} slot_{tool} f\left(\sum_{product=1}^{P} processing\_time_{product,tool,1} quantity_{product} \right) \leq capacity_1$$

$$\vdots$$

$$\sum_{tool=1}^{T} slot_{tool} f\left(\sum_{product=1}^{P} processing\_time_{product,tool,M} quantity_{product} \right) \leq capacity_M$$

$$\text{where } f(y) = \begin{cases} 1, & \text{if } y > 0 \\ 0, & \text{if } y = 0 \end{cases}$$

Exhibit C.

$$\sum_{product=1}^{P} \sum_{tool=1}^{T} processing\_time_{product,tool,1} quantity_{product} \leq available\_time_1$$

$$\vdots$$

$$\sum_{product=1}^{P} \sum_{tool=1}^{T} processing\_time_{product,tool,M} quantity_{product} \leq available\_time_M$$

$$quantity_{product} \geq 0,$$

$$quantity_{product} \leq Q_{product}, \text{ and}$$

$$quantity_{product} \text{ is an integer, for } product = 1, 2, \cdots, P$$

system. The way in which candidate solutions are encoded is one of a central factor in the success of GAs (Mitchell, 1996). Generally, the solution encoding can be defined over an alphabet Σ which might consist of binary digits, continuous numbers, integers, or symbols. However, choosing the best encoding scheme is almost tantamount to solving the problem itself (Mitchell, 1996). In this research, our GA system is mainly based on Holland's canonical model (Holland, 1992), which

is one of the most commonly used encoding schemes in practice—binary encoding.

A candidate solution for the batch selection task is a vector of quantities to be manufactured for *P* products. Let the entire solution space be denoted as ***solution*** (see Exhibit D).

The encoding function encodes the quantity to be produced for each product as an *l*-bit binary string, and then forms a concatenation of the strings for *P* products

Exhibit D.

$$\textbf{solution} = \prod_{product=1}^{P}[0, 1, \ldots, Q_{product}]$$

$$= \{(quantity_1, \ldots, quantity_P) \in (\{0\} \cup \aleph)^{P} \mid 0 \le quantity_{product} \le Q_{product},$$

$$quantity_{product} \text{ is an integer, and } product = 1, 2, \ldots, P\}$$

which are to be included in a production batch. Each candidate solution $(quantity_1, \ldots, quantity_P)$ is a string of length lP over the binary alphabet $\Sigma = \{0, 1\}$. Such an encoded l-bit string has a value equal to

$$\begin{cases} quantity_{product}, & \text{if } \max_{product=1,2,\ldots,P}\{Q_{product}\} \le 2^{l} - 1 \\ \left\lceil \dfrac{quantity_{product}(2^{l}-1)}{\max_{product=1,2,\ldots,P}\{Q_{product}\}} - 0.5 \right\rceil, & \text{otherwise.} \end{cases}$$

In the above formula, $2^{l} - 1$ is the value of an l-bit string $\underbrace{1 \ldots 1}_{l}$, and $\varphi \bullet \kappa$ is the ceiling function. For example, assume there are only two products to be selected in a production batch with 200 units as the largest possible quantity to be manufactured for each product. A candidate solution consisting of quantities 100 and 51 for products 1 and 2 respectively will be represented by a 16-bit string as 0110010000110011 with the first 8 bits representing product 1 and the second 8 bits representing product 2.

After a new solution string is generated, it is then decoded back to the format for the compuation of the objective function and for the check of solution feasibility. Let each l-bit segment of a solution string be denoted as *string* with *string*[i] as the value of the ith bit in the l-bit segment. The decoding function converts each l-bit string according to the following formula:

$$\begin{cases} \sum_{i=1}^{l} string[i] \cdot 2^{i-1}, & \text{if } \max_{product=1,2,\ldots,P}\{Q_{product}\} \le 2^{l} - 1 \\ \left\lceil \left(\sum_{i=1}^{l} string[i] \cdot 2^{i-1}\right) \cdot \dfrac{\max_{product=1,2,\ldots,P}\{Q_{product}\}}{2^{l}-1} - 0.5 \right\rceil, & \text{otherwise.} \end{cases}$$

Five-Phase Genetic Operations

Our system follows the generation-evaluation-selection-crossover-mutation cycles in searching for appropriate solution strings for the batch selection task. It starts with generating an initial population, *Pop*, of *pop_size* candidate solution strings at random. In each iteration of the operational cycle, each candidate solution string, s_i, in the current population is evaluated by the fitness function.

Candidate solution strings in the current population are selected probabilitistically on the basis of their fitness values as seeds for generating the next generation. The purpose of selection is to generate offspring of high fitness value on the basis of the fitter members in the current population. Actually, selection is the mechanism that helps our GA system to exploit a promising region in the solution space. There are several fitness-based schemes for the selection process: Roulette-wheel selection, rank-based selection, tournament selection, and elitist selection (Goldberg, 1989)(Michalewicz, 1994). The first three methods randomly select candidate solution strings for reproduction on the basis of either the fitness value or the rank of individual strings. Best members of the current population might be lost if they are not selected to reproduce or if they are altered by crossover (i.e., recombination) or mutation. The elitist selection strategy is for the purpose of retaining some of the fittest individuals from the current population.

Elitist selection retains a limited number of "elite" solution strings, i.e., strings with the best fitness

values, for passing to the next generation without any modification. A fraction called the "generation gap" is used to specify the proportion of the population to be replaced by offspring strings after each iteration. Our GA system retains copies of the first

$(1 - generation\_gap) \cdot pop\_size$ "elitist" members of **Pop** for the formation of the next population, **Pop**$_{new}$.

For generating the rest of the members for **Pop**$_{new}$, the GA module will probabilitistically select:

$$\frac{generation\_gap \cdot pop\_size}{2}$$

pairs of solution strings from **Pop** for generating offspring strings. The probability of selecting a solution string, s_i, from **Pop** is given by

$$\Pr(s_i) = \frac{Fitness(s_i)}{\sum_{j=1}^{pop\_size} Fitness(s_j)}.$$

Let the cumulative probability of individual solution strings in the population be called C_i, and

$$C_i = \sum_{j=1}^{i} \Pr(s_j),$$

for $i = 1, 2,\ldots, pop\_size$. The solution string s_i will be selected for reproduction if $C_{i-1} < rand(0,1) \leq C_i$.

In addition to exploiting a promising solution region via the selection process, we also need to explore other promising regions for possible better solutions. Exploitation without exploration will cause degeneration for a population of solution strings, and might cause the local optima problem for the system. Actually, the capability of maintaining a balanced exploitation vs. exploration is a major strength of the GA approach over traditional optimization techniques. The exploration function is achieved by the crossover and mutation operators. These two operators generate offspring solutions which belong to new schemata, and thus allow our system to explore other promising regions in a solution sapce. This process also allows our system to improve its performance stochastically.

Crossover recombines good solution strings in the current population and proliferates the population gradually with schemata of high fitness values. Crossover is commonly regarded as the most distinguishing operator of GAs, and it usually interacts in a highly intractable manner with fitness function, encoding, and other details of a GA (Mitchell, 1996). Though various crossover operators have been proposed, there is no general conclusions on when to use which type of crossover (Michalewicz, 1994)(Mitchell, 1996).

In this paper, we adopt the standard one-point crossover for our GA system. For each pair of solution strings selected for reproduction, the value of *crossover_rate* determines the probability for their recombination. A position in both candidate solution strings is randomly selected as the crossover point. The parts of two parent strings after the crossover position are exchanged to form two offspring. Let k be the crossover point randomly generated from a uniform distribution ranging from 1 to lP, where lP is the length of a solution string. Let $s_i = (x_1, x_2,\ldots, x_{k-1}, x_k,\ldots, x_{lP})$ and $s_j = (y_1, y_2,\ldots, y_{k-1}, y_k,\ldots, y_{lP})$ represent a pair of candidate solution strings selected for reproduction. Based on these two strings, the crossover operator generates two offspring

$s_i' = (x_1', x_2',\ldots, x_{lP}')$ and $s_j' = (y_1', y_2',\ldots, y_{lP}')$, where

$$x_i' = \begin{cases} x_i, & \text{if } i < k \\ y_i, & \text{otherwise} \end{cases}$$

$$y_i' = \begin{cases} y_i, & \text{if } i < k \\ x_i, & \text{otherwise.} \end{cases}$$

In other words, $s_i' = (x_1, x_2,\ldots, x_{k-1}, y_k,\ldots, y_{lP})$ and $s_j' = (y_1, y_2,\ldots, y_{k-1}, x_k,\ldots, x_{lP})$. These two oppspring are then added to **Pop**$_{new}$. This offspring-generating process is repeated until there are $generation\_gap \cdot pop\_size$ offspring generated for **Pop**$_{new}$.

With selection and crossover alone, our system might occasionally develop a uniform population which consists of the same solution strings. This will blind our system to other possible solutions. Mutation, which is the other operator applied to the reproduction process, is used to help our system avoid the formation of a uniform population by introducing diversity into a population. It is generally believed that mutation alone does not advance the search for a solution, and is usu-

ally considered as a secondary role in the operation of GAs (Goldberg, 1989). Usually, mutation is applied to alter the bit value of a string in a population only occasionally. Let *mutation_rate* be the probability of mutation for each bit in a candidate solution string.

For each offspring string, $s' = (x'_1, x'_2, \ldots, x'_{lP})$, generated by the crossover operator for the new population Pop_{new}, the mutation operator will invert each bit probabilitistically:

$$x'_i = \begin{cases} 1 - x_i, & \text{if } rand(0,1) < mutation\_rate \\ x_i, & \text{otherwise.} \end{cases}$$

The probability of mutation for a candidate solution string is $1 - (1 - mutation\_rate)^{lP}$.

The above processes constitute an operational cycle of our system. These operations are repeated until the termination criterion is reached, and the result is passed to the Decoder for decoding. The decoded result is then presented to the decision maker for further consideration in the final decision. If current solution is not satisfactory to the decision maker, the current solution can be modified by the decision maker, and then entered into the GA system to initiate another run of search process for satisfactory solutions.

FUTURE TRENDS AND CONCLUSION

In this paper we designed a GA-based system for the batch selection problem of flexible manufacturing systems. In our design we adopted a binary encoding scheme, the elitist selection strategy, a single-point crossover strategy, and a uniform random mutation for the batch selection problem.

The performance of GAs is usually influenced by various parameters and the complicated interactions among them, and there are several issues worth further investigation. With the availability of a larger pool of diverse schemata in a larger population, our GA system will have a broader view of the "landscape" (Holland, 1992) of the solution space, and is thus more likely to contain representative solutions from a large number of hyperplanes. This advantage gives GAs more chances of discovering better solutions in the solution space. However, Davis (1991) argues that the most effective population size is dependent upon the nature of the problem, the representation formalism, and the GA

operators. Still, Schaffer *et al*. (1991) asserted that the best settings for population size is independent of the problems. In the sequel of this paper, we will conduct a sequence of experiment to systematically analyze the influence of the population size on GA performance, by using the batch-selection model peoposed in this paper, so that we can be more conclusive on the issue of the effective population size.

REFERENCES

Arabas, J., & Kozdrowski, S. (2001). Applying an Evolutionary Algorithm to Telecommunication Network Design. *IEEE Transactions on Evolutionary Computation*. (5)4, 309-322.

Ballester, P.J., & Carter, J.N. (2004). An Effective Real-Parameter Genetic Algorithm with Parent Centric Normal Crossover for Multimodal Optimisation. *Proceedings of the 2004 GECCO. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer-Verlag. 901-913.

Davis, L. (Editor) (1991). *Handbook of Genetic Algorithms*. New York, NY: Van Nostrand Reinhold.

Del Rosso, C. (2006). Reducing Internal Fragmentation in Segregated Free Lists Using Genetic Algorithms. *Proceedings of the 2006 ACM Workshop on Interdisciplinary Software Engineering Research*.

Deng, P-S. (2007). A Study of the Performance Effect of Genetic Operators. *Encyclopedia of Artificial Intelligence*, Dopico, J.R.R., de la Calle, J.D. & Sierra, A.P. (Editors), Harrisburg, PA: IDEA.

Goldberg, D.E. (1989). *Genetic Algorithm in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

Gordon, M. (1988). Probabilistic and Genetic Algorithms for Document Retrieval. *Communications of the ACM*. (31)10, 1208-1218.

Grefenstette, J.J. (1993). Introduction to the Special Track on Genetic Algorithms. *IEEE Expert*. October, 5-8.

Holland, J. (1992). *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press.

Kean, J. (1995). Genetic Algorithms for Market Trading. *AI in Finance*. Winter, 25-29.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*. New York, NY: Springer-Verlag.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

Özdamar, L. (1999). A Genetic Algorithm Approach to a General Category Project Scheduling Problem. *IEEE Transactions on Systems, Man, and Cybernetics*. (29)1, 44-59.

Pierre, S., & Legault, G. (1998). A Genetic Algorithm for Designing Distributed Computer Network Topologies. *IEEE Transactions on Systems, Man, and Cybernetics*. (28) 249-258.

Reeves, C.R., & Rowe, J.E. (2003). *Genetic Algorithms - Principles and Perspectives*. Boston, MA: Kluwer Academic.

Richter, J.N., & Paxton, J. (2005). Adaptive Evolutionary Algorithms on Unitation, Royal Road and Longpath Functions. *Proceedings of the Fourth IASTED International Conference on Computational Intelligence*, Calgary, Alberta, Canada.

Shin, K.S., & Han, I. (1999). Case-Based Reasoning Supported by Genetic Algorithms for Corporate Bond Rating. *Expert Systems With Applications*. (16)2, 85-95.

Vafaie, H., & De Jong, K.A. (1998). Feature Space Transformation Using Genetic Algorithms. *IEEE Intelligent Systems*. (13)2, 57-65.

KEY TERMS

Batch Selection: Selecting the optimal set of products to produce, with each product requiring a set of resources, under the system capacity constraints

Fitness Functions: The objective function of the GA for evaluating a population of solutions

Flexible Manufacturing Systems: A manufacturing system which maintains the flexibility of order of operations and machine assignment in reacting to planned or unplanned changes in the production process

Genetic Algorithms: A stochastic search method which applies genetic operators to a population of solutions for progressively generating optimal or near-optimal solutions

Genetic Operators: Selection, crossover, and mutation, for combining and refining solutions in a population

Implicit Parallelism: A property of the GA which allows a schema to be matched by multiple candidate solutions simultaneously without even trying

Landscape: A function plot showing the state as the "location" and the objective function value as the "elevation"

Reinforcement Learning: A learning method which interprets feedback from an environment to learn optimal sets of condition/response relationships for problem solving within that environment

Schemata: A general pattern of bit strings that is made up of 1, 0, and #, used as a building block for solutions of the GA

Genetic Algorithms for Wireless Sensor Networks

João H. Kleinschmidt
State University of Campinas, Brazil

INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of low-cost and low-power sensor nodes. Some of the applications of sensor networks are environmental observation, monitoring disaster areas and so on. Distributed evolutionary computing is a poweful tool that can be applied to WSNs, because these networks require algorithms that are capable of learning independent of the operation of other nodes and also capable of using local information (Johnson, Teredesai & Saltarelli, 2005). Evolutionary algorithms must be designed for the resource constraints present in WSNs. This article describes how genetic algorithms can be used in WSNs design in order to satisfy energy conservation and connectivity constraints.

BACKGROUND

The recent advances in wireless communications and digital electronics led to the implementation of low power and low cost wireless sensors. A sensor node must have components for sensing, data processing and communication. These devices can be grouped to form a sensor network (Akyildiz, Sankarasubramaniam & Cayirci, 2002) (Callaway 2003). The network protocols, such as formation algorithms, routing and management, must have self-organizing capabilities. In

general, sensor networks have some features that differ from traditional wireless networks in some aspects: the number of sensor nodes can be very high; sensor nodes are prone to failures; sensor nodes are densely deployed; the topology of the network can change frequently; sensor nodes are limited in computational capacities, memory and energy.

The major challenge in the design of WSNs is the fact that energy resources are significantly more limited than in wired networks and other types of wireless networks. The battery of the sensors in the network may be difficult to recharge or replace, causing severe limitations in the communication and processing time between all sensors in the network. Thus, the main parameter to optimize for is the network lifetime, or the time until a group of sensors runs out of energy. Another issue in WSN design is the connectivity of the network according to the selected communication protocol. Usually, the protocol follows the cluster-based architecture, where single hop communication occurs between sensors of a cluster and a selected cluster head sensor that collects all information obtained by the other sensors in its cluster. This architecture is shown in Figure 1. Since the purpose of the sensor network is the collection and management of measured data for some particular application, this collection must meet specific requirements depending on the type of data. These requirements are turned into application specific parameters of the network.

Figure 1. Cluster-based sensor network
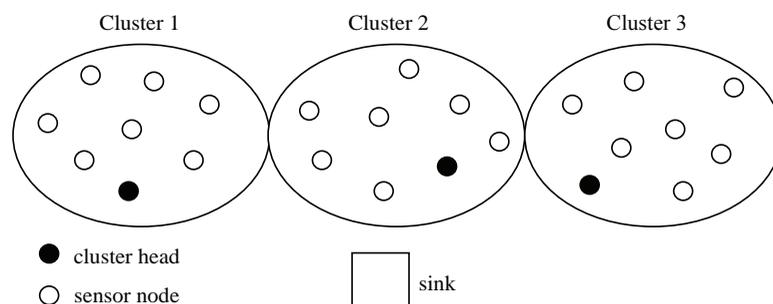
GENETIC ALGORITHMS FOR WIRELESS SENSOR NETWORKS

A WSN designer who takes into account all the design issues deals with more than one non-linear objective functions or design criteria which should be optimized simultaneously. Therefore, the focus of the problem is how to find many near-optimal non-dominated solutions in a practically acceptable computational time (Jourdan & de Weck, 2004) (Weise, 2006) (Ferentinos & Tsiligiridis, 2007). There are several interesting approaches to tackling such problems, but one of the most powerful heuristics, which is also appropriate to apply in the multi-objective optimization problem, is based on genetic algorithms (GA) (Ferentinos & Tsiligiridis, 2007).

Genetic algorithms have been used in many fields of science to derive solutions for any type of problems (Goldberg 1989) (Weise, 2006). They are particularly useful in applications involving design and optimization, where there are large numbers of variables and where procedural algorithms are either non-existent or extremely complicated (Khana, Liu & Chen, 2006), (Khana, Liu & Chen, 2007). In nature, a species adapts to an environment because the individuals that are the fittest in respect to that environment will have the best chance to reproduce, possibly creating even fitter child. This is the basic idea of genetic evolution. Genetic algorithms start with an initial population of random solution candidates, called individuals or chromosomes. In the case of sensor networks, the individuals are small programs that can be executed on sensor nodes (Wazed, Bari, Jaekel & Bandyopadhyay, 2007).

Each individual may be represented as a simple string or array of genes, which contain a part of the solution. The values of genes are called alleles. As in nature, the population will be refined step by step in a cycle of computing the fitness of its individuals, selecting the best individuals and creating a new generation derived from these. A fitness function is provided to assign the fitness value for each individual, based on how close an individual is to the optimal solution. Two randomly selected individuals, the parents, can exchange genetic information in a process called crossover to produce two new chromosomes know as child. A process called mutation may also be applied to obtain a good solution, after the process of crossover. This process helps to restore any genetic values when the population converges too fast. After the crossover and mutation processes the individuals of the next generation are selected. Some of the poorest individuals of the generation can be replaced by the best individuals from the previous generation. This is called elitism, and ensures that the new generation is at least as fit as the previous generation. The algorithm stops if a predetermined stopping criterion is met (Hussain, Matin & Islam, 2007).

Fitness Function and Specific Parameters for WSNs

The fitness function executed in a sensor node is a weighted function that measures the quality or performance of a solution, in this case a specific sensor network design. This function is maximized by the GA system in the process of evolutionary optimization. A fitness function must include and correctly represent all or at least the most important factors that affect the performance of the system. The major issue in developing a fitness function is the decision on which factors are the most important ones (Ferentinos & Tsiligiridis, 2007) (Gnanapandithan & Natarajan, 2006).

A genetic algorithm must be designed for WSN topologies by optimizing energy-related parameters that affect the battery consumption of the sensors and thus, the lifetime of the network. At the same time, the algorithm has to meet some connectivity constraints and optimize some physical parameters of the WSN implemented by the specific application. The multiple objectives of the optimization problem are blended into a single objective function, the parameters of which are combined to formulate a fitness function that gives a quality measure to each WSN topology. Three sets of parameters dominate the design and the performance of a WSN: the application specific parameters, connectivity parameters and the energy related parameters. Some possible parameters are discussed in (Ferentinos & Tsiligiridis, 2007):

- Operation energy: the energy that a sensor consumes during some specific time of operation. It depends whether the sensor operates as cluster head or as regular sensor.
- Communication energy: the energy consumption due to communication between sensors. It depends on the distances between transmitter and receiver.

- Battery life: battery capacity of each sensor.
- Sensors-per-cluster head: parameter to ensure that each cluster head does not have more than a maximum predefined number of sensors in its cluster. It depends on the physical communications capabilities and the amount of data that can be processed by a cluster head.
- Sensors out of range error: parameter to ensure that each sensor can communicate with its cluster head. It depends on the signal strength of the sensors.
- Spatial density: minimal number of measurements points that adequate monitor the variables of a given area.
- Uniformity of measurement: the measures of an area of interest must give a uniform view of the area conditions. The total area can be divided in several sub-areas for a uniform measurement.

Other parameters can be defined, especially those related to application specific requirements, such as sensor to sink delay, routing information, localization, network coverage, etc. The optimization problem is defined by the minimization of the WSN parameters. If *n* optimization parameters were defined, they may be combined into a single objective function:

$$f = \min\left\{\sum_{i=1}^{n} w_i P_i\right\},$$

where *P* is the parameter objective and *w* is the weighting coefficients, that define the importance of each parameter in the network design. The importance of each parameter on the performance of the network has to be designed carefully. These values are firstly determined based on experience on the importance of each one. Then, some experimentation is made to determine the final values. An individual will be selected to be the parent of the next generation using its fitness value. The probability that an individual be chosen is proportional to the value. After this process, the type of crossover and mutation has to be defined, as well as the population size and the probabilities for crossover and mutation. Some experiments must be carried out to determine the most appropriate values for WSNs.

FUTURE TRENDS

Some of the recent research areas in wireless sensor networks include the design of MAC protocols, efficient routing, data aggregation, collaborative processing, sensor fusion, security, localization, data reliability, network management, etc. All these topics may benefit from the usage of genetic algorithms. Some research has been made using genetic algorithms to solve some WSNs problems (Hussain, Matin & Islam, 2007) (Jin, Liu, Hsu & Kao, 2005) (Ferentinos & Tsiligiridis, 2007) (Wazed, Bari, Jaekel & Bandyopadhyay, 2007) (Rahmani, Fakhraie, & Kamarei, 2006) (Qiu, Wu, Burns, & Holzhauer, 2006). However, most of the research topics of WSNs using genetic algorithms remain few or completely unexplored.

CONCLUSION

This article discussed the application of genetic algorithms in wireless sensor networks. The basic idea of GA was discussed and some specific considerations for WSNs were made, including crossover, mutation and definition of the fitness function. The mainly performance parameters may be divided in three groups: energy, connectivity and application specific. Since WSNs have many objectives to be optimised, GA is a promising candidate to be used in WSNs design.

REFERENCES

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine,* 40 (8), 102-114.

Callaway, Egdar H. (2003). Wireless Sensor Networks: Architectures and Protocols, CRC Press, 352 pages.

Ferentinos, K. P., & Tsiligiridis, T. A. (2007). Adaptive Design Optimization of Wireless Sensor Networks Using Genetic Algorithms. *Elsevier Computer Networks,* (51) 1031-1051.

Gnanapandithan, N. & Natarajan, B. (2006). Parallel Genetic Algorithm Based Optimal Fusion in Sensor Networks, IEEE Consumer Communications and Networking Conference.

G

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA, 1989.

Hussain, S., Matin, A. W., & Islam, O. (2007). Genetic Algorithm for Energy Efficient Clusters in Wireless Sensor Networks. IEEE 4th International Conference on Information Technology, Las Vegas, Nevada, USA.

Jin, M., Liu, W., Hsu, D. F., & Kao, C. (2005). Compact Genetic Algorithm for Performance Improvement in Hierarchical Sensor Networks Management. IEEE Int. Symposium on Parallel Architectures, Algorithms and Networks, Las Vegas, USA.

Johnson, D., Teredesai, A. M., & Saltarelli, R. (2005). Genetic Programming in Wireless Sensor Networks. European Conference on Genetic Programming, Lausanne, Switzerland.

Jourdan, D. B. & de Weck, O. L. (2004). Layout Optimization for a Wireless Sensor Network Using a Multi-objective Genetic Algorithm. IEEE Vehicular Technology Conference.

Khana, R., Liu, H., & Chen, H. (2006). Self-Organization of Sensor Networks Using Genetic Algorithms. IEEE International Conference on Communications, Istanbul, Turkey.

Khana, R., Liu, H., & Chen, H. (2007). Dynamic Optimization of Secure Mobile Sensor Networks: A Genetic Algorithm. IEEE International Conference on Communications, Glasgow, Scotland.

Qiu, Q., Wu, Q., Burns, D. & Holzhauer, D. (2006). Lifetime Aware Resource Management for Sensor Network Using Distributed Genetic Algorithm. International Symposium on Low Power Electronics and Design.

Rahmani, E., Fakhraie, S. M. & Kamarei, M. (2006). Finding Agent-Based Energy-Efficient Routing in Sensor Networks using Parallel Genetic Algorithm, International Conference on Microelectronics.

Wazed, S., Bari, A., Jaekel, A., & Bandyopadhyay, S. (2007). Genetic Algorithm Based Approach for Extending the Lifetime of Two-Tiered Sensor Networks. 2nd IEEE International Symposium on Wireless Pervasive Computing, San Juan, Puerto Rico.

Weise, T. Genetic Programming for Sensor Networks. (2006) Technical report, University of Kassel.

KEY TERMS

Cluster-Based Architecture: Sensor networks architecture where communication occurs between sensors of a cluster and a selected cluster head that collects the information obtained by the sensors in its cluster.

Cluster Head: Sensor node responsible for gathering data of a sensor cluster and transmitting them to the sink node.

Crossover: Genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next.

Energy Parameters: Parameters that affect the battery consumption of the sensors, including the energy consumed due to sensing, communication and computational tasks.

Fitness Function: A particular type of objective function that quantifies the optimality of a solution in a genetic algorithm.

Genetic Algorithms: Search technique used in computing to find true or approximate solutions to optimization and search problems.

Mutation: The occasional (low probability) alteration of a bit position.

Network Lifetime: Time until the first sensor node or group of sensor nodes in the network runs out of energy.

Sensor Node: Network node with components for sensing, data processing and communication.

Wireless Sensor Networks: A network of spatially distributed devices using sensors to monitor conditions at different locations, such as temperature, sound, pressure, etc.

Genetic Fuzzy Systems Applied to Ports and Coasts Engineering

Óscar Ibáñez
University of A Coruña, Spain

Alberte Castro
University of Santiago de Compostela, Spain

INTRODUCTION

Fuzzy Logic (FL) and fuzzy sets in a wide interpretation of FL (in terms in which fuzzy logic is coextensive with the theory of fuzzy sets, that is, classes of objects in which the transition from membership to non membership is gradual rather than abrupt) have placed modelling into a new and broader perspective by providing innovative tools to cope with complex and ill-defined systems. The area of fuzzy sets has emerged following some pioneering works of Zadeh (Zadeh, 1965 and 1973) where the first fundamentals of fuzzy systems were established.

Rule based systems have been successfully used to model human problem-solving activity and adaptive behaviour. The conventional approaches to knowledge representation are based on bivalent logic. A serious shortcoming of such approaches is their inability to come to grips with the issue of uncertainty and imprecision. As a consequence, the conventional approaches do not provide an adequate model for modes of reasoning. Unfortunately, all commonsense reasoning falls into this category.

The application of FL to rule based systems leads us to fuzzy systems. The main role of fuzzy sets is representing Knowledge about the problem or to model the interactions and relationships among the system variables. There are two essential advantages for the design of rule-based systems with fuzzy sets and logic:

- The key features of knowledge captured by fuzzy sets involve handling uncertainty.
- Inference methods become more robust and flexible with approximate reasoning methods of fuzzy logic.

Genetic Algorithms (GAS) are a stochastic optimization technique that mimics natural selection (Holland, 1975). GAs are intrinsically robust and capable of determining a near global optimal solution. The use of GAS is usually recommended for optimization in high-dimensional, multimodal complex search spaces where deterministic methods normally fail. GAs explore a population of solutions in parallel. The GA is a searching process based on the laws of natural selections and

Figure 1. A typical GA cycle
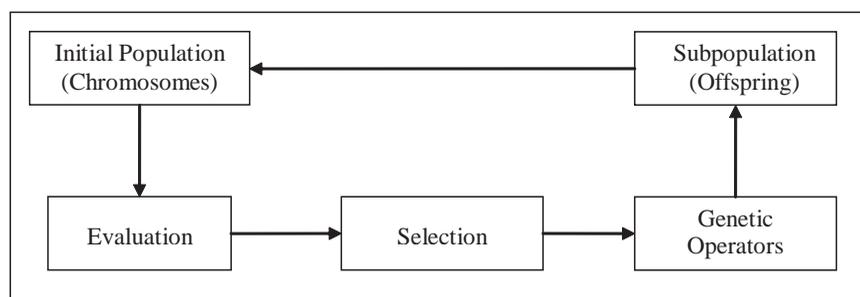
genetics. Generally, a simple GA contains three basic operations: selection, genetic operations and replacement. A typical GA cycle is shown in Fig. 1.

In this paper it is shown how a genetic algorithm can be used in order to optimize a fuzzy system which is used in wave reflection analysis at submerged breakwaters.

BACKGROUND

Many works have been done in the area of artificial intelligence applied to Coastal Engineering. It can be said that Artificial Intelligence methods have a wide acceptance among Coastal & Ports Engineers. Artificial Neural Network has been applied for years with very good results. The big drawback is their inability to explain their results, how have reached them, because they work as a black box and it can not be known what happen inside them. Over the last few years, a lot of works about fuzzy systems with engineering applications have been developed (Mercan, Yagci & Kabdasli, 2003; Dingerson, 2005; Gezer, 2004; Ross, 2004; Oliveira, Souza & Mandorino, 2006; Ergin, Williams & Micallef, 2006; Yagci, Mercan, Cigizoglu & Kabdasli, 2005). These systems have the advantage of being easy to understand (their solutions) and the capacity to handle uncertainty. However, most of these found a problem with knowledge extraction; when they try to define their RB and DB, in many cases for the difficulty of the problem and more often for the difficulty of represent all the expert knowledge in some rules and membership function.

To overcome these problems Genetic Fuzzy Systems (GFS) emerged, in which expert advice it is not as important as in Fuzzy System (FS) since it could be only needed to define the variables involved and its work domain. GFS (Cordón, et al., 2001) allow us to be less dependent on expert knowledge and in addition it is easier to reach better accuracy with these systems since they can realize a tuning process for membership functions and refine the rule set in order to optimize it. Following a specific application of GFS for wave reflection analysis at submerged breakwaters is presented.

While other kinds of techniques have been applied to that problem (Taveira, 2005; Kobayasi & Wurjanto, 1989; Abul-Azm, 1993; Losada, Silva & Losada, 1999),

it is a novel approach to estimate reflection coefficient, since a GA will determine the membership functions for each variable involved in the fuzzy system.

ANALYSIS OF WAVE REFLECTION AT SUBMERGED BREAKWATERS WITH A GENETIC FUZZY SYSTEM

Fuzzy rule-based systems can be used as a tool for modelling non-linear systems especially complex physical systems. It is well known fact that the breakwater damage ratio estimation process is dynamic and non-linear, so classical methods cannot be able to capture this behaviour resulting in unsatisfactory solutions.

The Knowledge Base (KB) is the FS component comprising the expert knowledge knows about the problem. So is the only component of the FS depending on the concrete application and it makes the accuracy of the FS depends directly on its composition. The KB is comprised of two components, a Data Base (DB), containing the definitions of fuzzy rules linguistic labels, that is, the membership functions of the fuzzy sets, and a Rule Base (RB), constituted by the collection of fuzzy rules representing the expert knowledge.

There are many tasks that have to be performed in order to design a concrete FS. As it has been shown previously, the derivation of the KB is the only one directly depending on the problem to solve. It is known that the more used method in order to perform this task is based directly on extracting the expert experience from the human process operator. The problem arises when there are not able to express their knowledge in terms of fuzzy rules. In order to avoid this drawback, researches have been investigating automatic learning methods for designing FSs by deriving automatically an appropriate KB for the FS without necessary of its human expert.

The Genetic algorithms (GA) have demonstrated to be a powerful tool for automating the definition of the KB since adaptativa control, learning and self-organization can be considered in a lot of cases as optimization or search process. The fuzzy systems making use of GA in their design process are called generically GFSs.

These advantages have extended the use of GAs in the development of a wide range of approaches for designing FSs in the last years. It is possible to

distinguish three different groups of genetic FS design process according to the KB components included in the learning process. These ones are the following:

- Genetic definition of the Fuzzy System Data Base (Bolata and Nowé, 1995; Fathi-Torbaghan and Hildebrand, 1994; Herrera and Verdegay, 1995b; Karr, 1991b).
- Genetic derivation of the Fuzzy System Rule Base (Bonarini, 1993; Karr, 1991a; Thrift, 1991).
- Genetic learning of the Fuzzy System Knowledge Base (Cooper and Vidal, 1993; Herrera, Lozano and Verdegay, 1995a; Leitch and Probert, 1994; Lee and Takagi, 1993; Ng and Lee, 1994).

In this paper, we create a Fuzzy System which predicts reflection coefficient at a different model of submerged breakwaters. To do this task, a part of this Fuzzy System, the Data Base, is defined and tuning by a Genetic Algorithm.

SUBMERGED BREAKWATER DOMAIN

Submerged breakwaters are effective shore protection structure against wave action with a reduced visual impact (see fig. 2).

To predict reflection coefficient several parameters have to be taken into account, they are:

- Rc: water level above crest.

- Hs: significant wave height.
- d: water depth.
- Tp: peak period or
- Lp: peak wavelength

These are parameters that connect the submerged breakwater model and the wave. The parameters that identified the submerged breakwater model (see fig. 3) are: the height (h) and the crest width (B), n (cotangent α), breakwater slope (α) and slope nature (smooth or rough). To predict the reflection coefficient, the first ones were used but in many cases dimensionless parameters were used instead the parameters separately.

A lot of tests were done with different number of input variables and different number of fuzzy sets for each membership function. Depending of the variables and membership function number, a set of rules were established for each case.

PHYSICAL TEST

A large number of tests have been carried out (Taveira-Pinto, 2001) with different water deeps and wave conditions for each model (figure 3 shows the general layout of the tested models). Eight impermeable physical models have been tested with different geometries (crest width, slope), different slope nature (smooth, rough), values for tan α (from 0.20 to 1.00) and n (from 1 to 5) in the old unidirectional wave tank of the Hydraulics Laboratory of the Faculty of Engineering of the University of Porto.

Figure 2. Outline of a submerged breakwater and its action
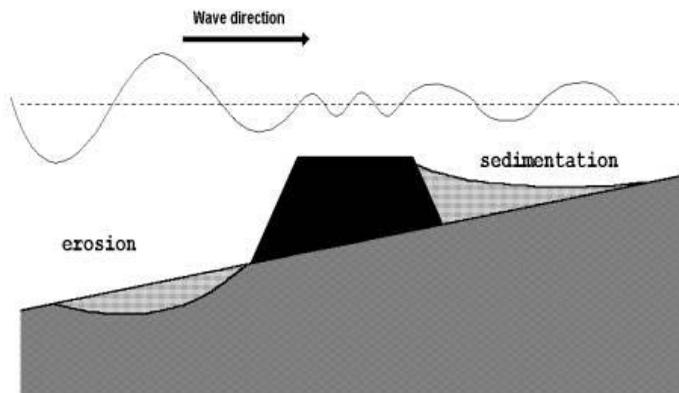
Figure 3. Diagram of interesting variables taken into account in a submerged breakwater
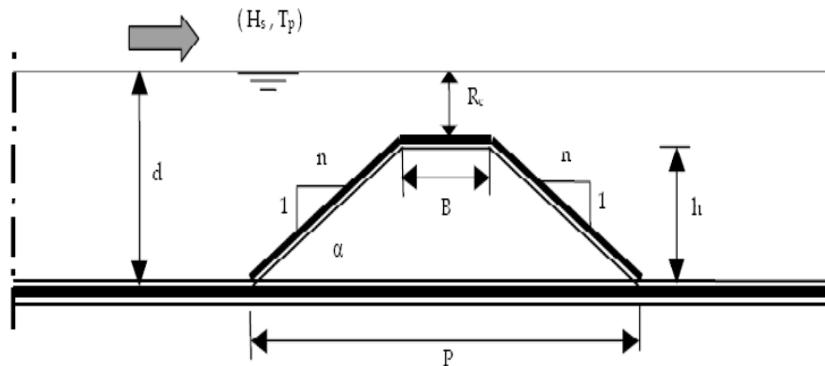
GENETIC FUZZY SYSTEM

The target of the GA is find the better distribution for the membership functions (optimization task) inside of the domain of each variable, so that minimizes the error of the created fuzzy system when it is applied to the training set

Genome Encoding

Each individual of the GA represents the Data Base of the fuzzy system that means all the membership functions. Each gen contains the position of one point of one membership function. As can be seen in fig. 4, one variable X with all its fuzzy sets is coding as a chain of real numbers.

The used codification allows different kinds of membership functions (triangular, trapezoid, Gaussian, etc…) codifying the representative points in the chromosome so the resultant chromosome is variable size.

Genetic Operators

Genetic operators were limited in order to generate meaningful fuzzy systems.

- **Crossover:** The classical crossover operator, with one-point, n-point or uniform crossover, has to be limited in its possible cross points. To avoid meaningless membership functions it is only allows exchange the genetic material corresponding to whole variables.
- **Mutation:** When a mutation happens, the new value of the gen will be between a lower and an upper limit, both have worked out from the neighbour points of the corresponding membership function and its neighbour membership functions.
- **Selection:** The selection method is tournament with elitism (Blickle, 1997).

Fitness

The way of find out what individual is better than other is the fitness function. In this case, one individual represent a part of a fuzzy system (DB) and with the rest of the fuzzy system (static RB) the fitness of that individual can be calculate. For that aim the physical test is split in two new sets, one was used as a training set and the other as a test set. For each physical test of the training set, the corresponding value for the input variables are introducing in the fuzzy system (individual in the genetic population). Once is calculated the output with a Mandani (Mandani, 1977) strategy and a Centroid defuzzification method, the result is compared to the output of the physical test; the difference is piled up for every tests in the training set and once all test have been introduced in the fuzzy system (one individual from the GA) and have been calculated its error, the

Figure 4. Piece of a chromosome. X_{ij} contains the position of one point (i) of one membership function (j)
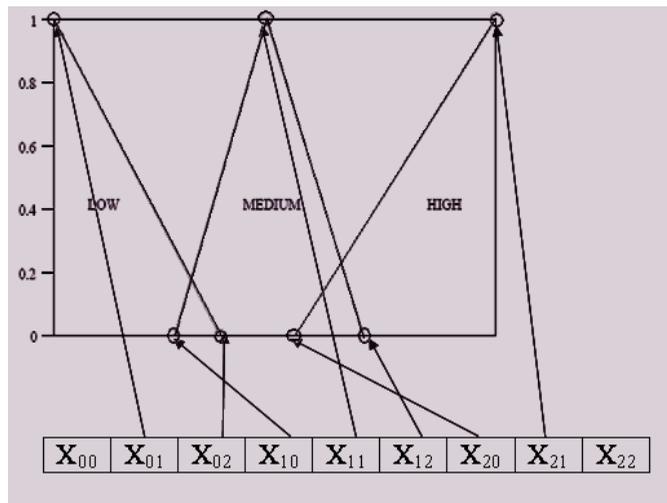
addition of the errors is the fitness function value for the individual. The smaller is the total error the better is the individual.

Results

Good results were obtained (from 85% to 95% of success) for the different tests done. Tests differ from one another for the number of input variables and the number of rules as well as genetic algorithm parameters. An easy understanding test is explained following:

- Selected dimensionless parameters: Rc/Hs and d/Lp.
- Both input variables were split in two (Low and High) trapezoidal membership functions.
- The output variable Cr (reflection coefficient) was split in three (Low, Medium and High) trapezoidal membership functions.
- The rule set was made up of by three rules:
  - o If (Rc/Hs = Low) and (d/Lp = Low) then (Cr = High)
  - o If (Rc/Hs = Low) and (d/Lp = High) then (Cr = Medium)
  - o If (Rc/Hs = High) and (d/Lp = Low) then (Cr = Medium)

The training set was made up of 24 physical tests and the medium square error in that step was 0.84. Resultant membership functions can be seen in fig. 5. The test set was made up of 11 physical tests and the mean square error in that step was 0.89.

FUTURE TRENDS

Give the GA the capacity to optimize rules so that the system definition becomes easier and better results can be reached. The GA must be able to generate individuals with different number rules and different kind of rules at the same time that these individuals represent different membership functions.

CONCLUSION

- A Genetic Fuzzy System was development to estimate the wave reflection coefficient at submerged breakwaters.
- Good results were obtained (near to 90% accuracy) but better results (near to 97% accuracy) are difficult to understand inside the fuzzy theory.

Figure 5. Resultant membership functions from tuning process of a DB by GA
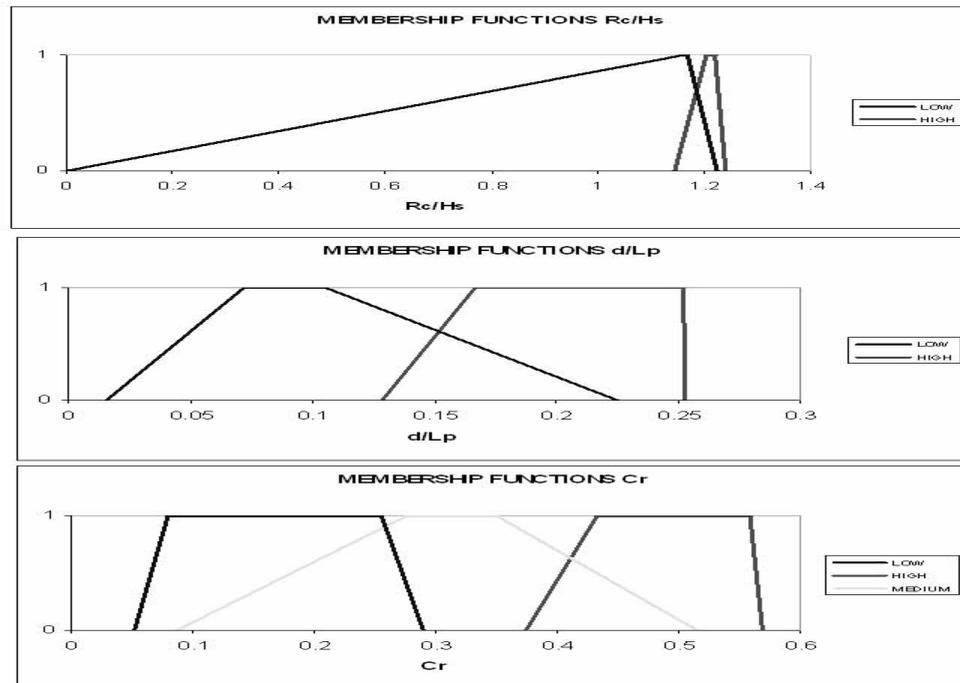
- It is a hard task to choose the rule set and furthermore the system's accuracy depends on this set a lot.
- The more inputs the problem have the more difficult become to define the rule set.

REFERENCES

Abul-Azm A. G., 1993. *Wave Diffraction Through Submerged Breakwaters*. Journal of Waterway, Port, Coastal and Ocean Engineering, Vol. 119, No. 6, pp. 587-605.

Baglio S. & Foti E., 2003. *Non-invasive measurements to analyze sandy bed evolution under sea waves action*. Instrumentation and Measurement, IEEE Transactions on. Vol. 52, Issue: 3, pp. 762-770.

Blickle, T. (1997). Tournament selection. In T. Bäck, D.G. Fogel, & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation.* New York: Taylor & Francis Group.

Bolata F. & Nowé A., 1995. *From fuzzy linguistic specifications to fuzzy controllers using evolution strategies*. In Proc. Fourth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'95), Yokohama, pp. 77-82.

Bonarini A. 1993. *Learning incomplete fuzzy rule sets for an autonomous robot*. In Proc. First European Congress on Fuzzy and Intelligent Technologies (EU-FIT'93), Aachen, pages 69-75.

Cooper M. G. & Vidal J. J., 1993. *Genetic design of fuzzy logic controllers.* In Proc. Second International Conference on Fuzzy Theory and Technology (FTT'93), Durham.

Cordón, O., Herrera, F., Hoffman F., Magdalena, L. (2001). *Genetic fuzzy systems.* World Scientific.

Dingerson L. M., 2005. *Predicting future shoreline condition based on land use trends, logistic regression and fuzzy logic.* Thesis. The Faculty of the School of Marine Science.

Ergin A., Williams A.T. & Micallef A., 2006. *Coastal Scenery: Appreciation and Evaluation.* Journal of

Coastal Research Article: pp. 958-964. Volume 22, Issue 4.

Fathi-Torbaghan M. & Hildebrand L., 1994. *Evolutionary strategies for the optimization of fuzzy rules*. In Proc. Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU'94), Paris, pp. 671-674.

Gezer E., 2004. *Coastal Scenic Evaluation, A pilot study for Çiralli.* Thesis. The Graduate School of Natural and Applied Sciences of Middle East Technical University.

Herrera F., Lozano M. & Verdegay J. L., 1995a. *A Learning process for fuzzy control rule using genetic algorithms.* Technical Report DECSAI-95108, University of Granada, Department of Computer Science and Artificial Intelligence.

Herrera F., Lozano M. & Verdegay J. L., 1995b. *Tuning fuzzy logic controllers by genetic algorithms.* International Journal of Approximate Reasoning 12: 293-315.

Holland J. H., 1975. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor.

Karr C., 1991a. *Applying genetics.* AI Expert pages 38-43.

Karr C., 1991b. *Genetics algorithms for fuzzy controllers.* AI Expert pages 26-33.

Kobayashi N. & Wurjanto A., 1989. *Wave Transmission Over Submerged Breakwaters.* Journal of Waterway, Port, Coastal and Ocean Engineering, Vol. 115, No. 5, pp. 662-680.

Leitch D. & Probert P., 1994. *Context depending coding in genetic algorithms for the sesign of fuzzy systems.* In Proc. IEEE/Nagoya University WWW on Fuzzy Logic and Neural Networks/Genetic Algorithms Nagoya.

Lee M. & Takagi H., 1993. *Embedding a priori knowledge into an integrate fuzzy system design method based on genetic algorithms.* In Proc. Fifth International Fuzzy Systems Association World Congress (IFSA'93), Seoul, pages 1293-1296.

Losada I.J., Silva R. & Losada, M.A., 1996. *3-D nonbreaking regular wave interaction with submerged breakwaters.* Coastal Engineering, Volume 28, Number 1, pp. 229-248(20).

Mandani, E.H., 1977. *Application of fuzzy logic to approximate reasoning using linguistic synthesis*. IEEE Transactions on Computers C-26 (12), 1182–1191.

Oliveira S.S., Souza F.J. & Mandorino F., 2006. *Using fuzzy logic model for the selection and priorization of port areas to hydrographic re-surveying.* Evolutions in hydrography, Antwerp (Belgium), Proceedings of the 15th International Congress of the International Federation of Hydrographic Societies. Special Publication of the Hydrographic Society, 55: pp. 63-67.

Ross T. J., 2004. *Fuzzy Logic with Engineering Applications*. John Wiley and Sons. Technology & Industrial Arts.

Taveira-Pinto F., 2001, *"Analysis of the oscillations and velocity fields in the vicinity of submerged breakwaters under wave action",* Ph. D. Thesis, Univ. of Porto (In Portuguese).

Taveira-Pinto F., 2005. *Regular water wave measurements near submerged breakwaters.* Meas. Science and Technology.

Thrift P., 1991. *Fuzzy logic synthesis with genetic algorithms*. In Proc. Fourth International Conference on Genetic Algorithms (ACGA'91), pages 509-513.

Van Oosten R. P., Peixó J., Van der Meer M.R.A. & Van Gent H.J., 2006. *Wave transmisión at low-crested structures using neural networks.* ICCE 2006, Abstract number: 1071.

Yagci O., Mercan D. E., Cigizoglu H. K. & Kabdasli M. S., 2005. *Artificial intelligence methods in breakwater damage ratio estimation*. Ocean engineering, vol. 32, no. 17-18, pp. 2088-2106.

Yagci O., Mercan D. E. & Kabdasli M. S., 2003. *Modelling Of Anticipated Damage Ratio On Breakwaters Using Fuzzy Logic*. EGS - AGU - EUG Joint Assembly, Abstracts from the meeting held in Nice, France, 6 - 11 April 2003.

Zadeh L. A., 1965. *Fuzzy sets*. Information and Control 8: 358-353.

Zadeh L. A., 1973. *Outline of a new approach to the analysis of complex systems and decision process*. IEEE Transactions on Systems, Man and Cybernetics 3: 28-44.

KEY TERMS

Fuzzification: Establishes a mapping from crisp input values to fuzzy set defined in the universe of discourse of that input.

Fuzzy System (FS): Any FL-based system, which either uses FL as the basis for the representation of different forms of knowledge, or to model the interactions and relationships among the system variables.

Genetic Algorithm: General-purpose search algorithms that use principles by natural population genetics to evolve solutions to problems

Genetic Fuzzy System: A fuzzy system that is augmented with an evolutionary learning process.

Mamdani Fuzzy Rule-Based System: A rule based system where fuzzy logic (FL) is used as a tool for representing different forms of knowledge about the problem at hand, as well as for modelling the interactions and relationships that exist between its variables.

Mamdani Inference System: Derives the fuzzy outputs from the inputs fuzzy sets according to the relation defined through fuzzy rules. Establishes a mapping between fuzzy sets $U = U_1$ x U_2 x . . . x U_n in the input domain of X_1…, X_n and fuzzy sets V in the output domain of Y. The fuzzy inference scheme employs the generalized modus ponens, an extension to the classical modus ponens (Zadeh, 1973).

Takagi-Sugeno-Kang Fuzzy Rule-Based System: A rule based system whose antecedent is composed of linguistic variables and the consequent is represented by a function of the input variables.

Grammar–Guided Genetic Programming

Daniel Manrique
Inteligencia Artificial, Facultad de Informatica, UPM, Spain

Juan Ríos
Inteligencia Artificial, Facultad de Informatica, UPM, Spain

Alfonso Rodríguez-Patón
Inteligencia Artificial, Facultad de Informatica, UPM, Spain

INTRODUCTION

Evolutionary computation (EC) is the study of computational systems that borrow ideas from and are inspired by natural evolution and adaptation (Yao & Xu, 2006, pp. 1-18). EC covers a number of techniques based on evolutionary processes and natural selection: evolutionary strategies, genetic algorithms and genetic programming (Keedwell & Narayanan, 2005).

Evolutionary strategies are an approach for efficiently solving certain continuous problems, yielding good results for some parametric problems in real domains. Compared with genetic algorithms, evolutionary strategies run more exploratory searches and are a good option when applied to relatively unknown parametric problems.

Genetic algorithms emulate the evolutionary process that takes place in nature. Individuals compete for survival by adapting as best they can to the environmental conditions. Crossovers between individuals, mutations and deaths are all part of this process of adaptation. By substituting the natural environment for the problem to be solved, we get a computationally cheap method that is capable of dealing with any problem, provided we know how to determine individuals' fitness (Manrique, 2001).

Genetic programming is an extension of genetic algorithms (Couchet, Manrique, Ríos & Rodríguez-Patón, 2006). Its aim is to build computer programs that are not expressly designed and programmed by a human being. It can be said to be an optimization technique whose search space is composed of all possible computer programs for solving a particular problem. Genetic programming's key advantage over genetic algorithms is that it can handle individuals (computer programs) of different lengths.

Grammar-guided genetic programming (GGGP) is an extension of traditional GP systems (Whigham, 1995, pp. 33-41). The difference lies in the fact that they employ context-free grammars (CFG) that generate all the possible solutions to a given problem as sentences, establishing this way the formal definition of the syntactic problem constraints, and use the derivation trees for each sentence to encode these solutions (Dounias, Tsakonas, Jantzen, Axer, Bjerregard & von Keyserlingk, D. 2002, pp. 494-500). The use of this type of syntactic formalisms helps to solve the so-called closure problem (Whigham, 1996). To achieve closure valid individuals (points that belong to the search space) should always be generated. As the generation of invalid individuals slows down convergence speed a great deal, solving this problem will very much improve the GP search capability. The basic operator directly affecting the closure problem is crossover: crossing two (or any) valid individuals should generate a valid offspring. Similarly, this is the operator that has the biggest impact on the process of convergence towards the optimum solution. Therefore, this article reviews the most important crossover operators employed in GP and GGGP, highlighting the weaknesses existing nowadays in this area of research. We also propose a GGGP system. This system incorporates the original idea of employing ambiguous CFG to overcome these weaknesses, thereby increasing convergence speed and reducing the likelihood of trapping in local optima. Comparative results are shown to empirically corroborate our claims.

BACKGROUND

Koza defined one of the first major crossover operators (KX) (1992). This approach randomly swaps subtrees in both parents to generate offspring. Therefore, it tends to disaggregate the so-called building blocks across the trees (that represent the individuals). The building blocks are those subtrees that improve fitness. This over-expansion has a negative effect on the fitness of the individuals. Also, this operator's excessive exploration capability leads to another weakness: an increase in the size of individuals, which affects system performance, and results in a lower convergence speed (Terrio & Heywood, 2002). This effect is known as bloat or code bloat.

There is another important drawback: many of the generated offspring are syntactically invalid as the crossovers are done completely at random. These individuals should not be part of the new population because they do not provide a valid solution. This seriously undermines the convergence process. Figure 1 shows a situation where one of the two individuals generated after Koza's crossover breaches the constraints established by a hypothetical grammar whose sentences represent arithmetic equalities.

The strong context preservative crossover operator (SCPC) avoids the problem of desegregation of building blocks (also called context) across the trees by setting severe (strong) constraints for tree nodes considered as possible candidates for selection as crossover nodes (D'haesler, 1994, pp. 379-407). A system of coordinates is defined to univocally identify each node in a derivation tree. The position of each node within the tree is specified along the path that must be followed to reach a given node from the root. To do this, the position of a node is described by means of a tuple of n coordinates $T = (b_1, b_{2,...,} b_n)$, where n is the node's depth in the tree, and b_i indicates which branch is selected at depth i (counting from left to right). Figure 2 shows an example representing this system of coordinates.

Only nodes with the same coordinates from both parents can be swapped. For this reason, a subtree may possibly never migrate to another place in the tree. This limitation can cause serious search space exploration problems, as the whole search space cannot be covered unless each function and terminal appears at every possible coordinate at least once in any one individual in the population. This failure to migrate building blocks causes them to evolve separately in each region, causing a too big an exploitation capability, thereby increasing the likelihood of trapping in local optima (Barrios, Carrascal, Manrique & Ríos, 2003, pp. 275-293).

As time moves on, the code bloat phenomenon becomes a serious problem and takes an ever more prominent role. To avoid this, Crawford-Marks &

Figure 1. Incorrect operation of Koza's crossover operator
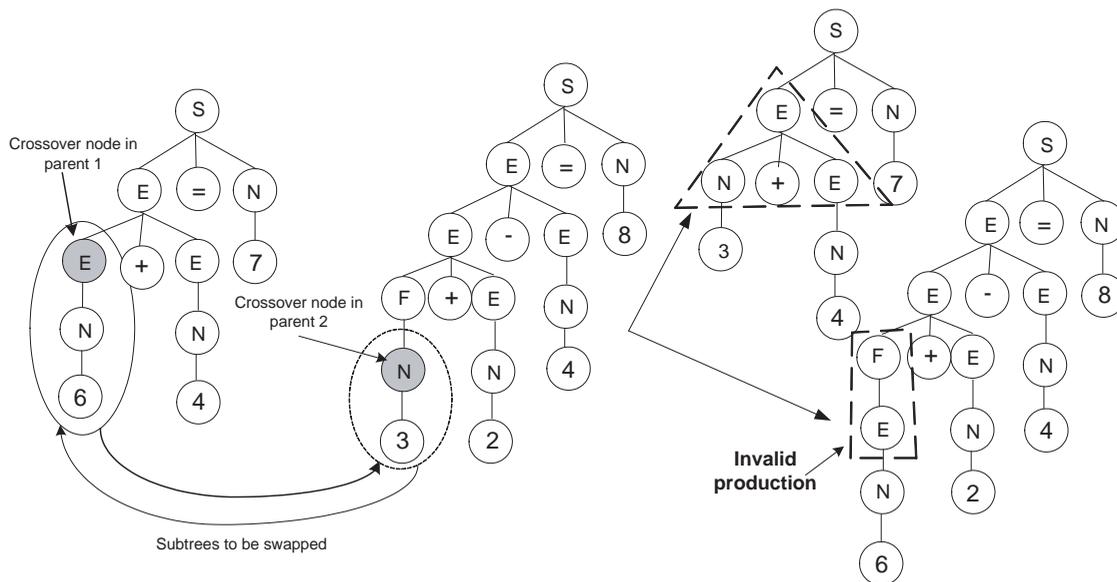
Figure 2. The system of coordinates defined in SCPC
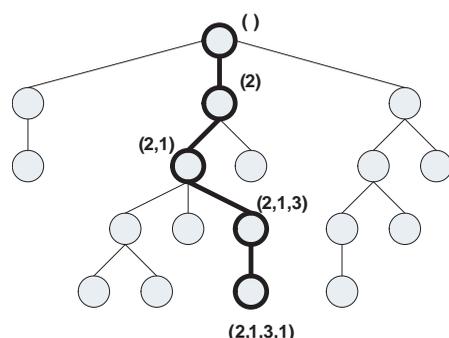
Spector (2002) developed the Fair crossover (pp. 733-739). This is a modified version of the approach proposed by Langdon (1999, pp. 1092-1097). Tree size is controlled as follows. First, a crossover node in the first parent is selected at random and the length, l, of the subtree extending from the node to the leaves is calculated. Then, a node is also selected at random in the second parent, and the length, l_2, for this second subtree is calculated. If l_2 is within the range $[l - l/4, l + l/4]$, then the crossover node for the second parent is accepted, and the two subtrees are swapped. If not, another crossover node is selected at random for the second parent and the check is run again. This way, the size of the subtree in the second parent to be swapped is controlled and limited, so the code bloat phenomenon is avoided. Another aspect to comment here is that the range in which l_2 must be included can be modified to afford specific problems more efficiently, but the range originally proposed works fine for most of them.

Whigham proposed one of the most commonly used operators (WX) in GGGP (1995, pp. 33-41). Because of its sound performance in such systems, it has become the de facto standard and is still in use today (Rodrigues & Pozo, 2002, pp. 324-333), (Hussain, 2003), (Grosman & Lewin, 2004, pp. 2779-2790). The algorithm works as follows. First, as all the terminal symbols have at least one non-terminal symbol above them, then, without loss of generality, the crossover nodes can be confined exclusively to locations on nodes containing non-terminal symbols. A non-terminal node belonging to the first parent is selected at random. Then a non-terminal node labeled with the same non-terminal symbol as in the first-chosen crossover node is selected from the second parent. This assures that generated individuals belong

to the grammar-generated language, as the crossed nodes share the same symbol. This operator's main flaw is that there are other possible choices of node in the second parent that are not explored and that could end in the target solution (Manrique, Marquez, Ríos & Rodríquez-Patón, 2005, pp. 252-261).

THE PROPOSED CROSSOVER OPERATOR FOR GGGP SYSTEMS

The proposed operator is a general-purpose operator designed to work in any GGGP system. It takes advantage of the key feature that defines a CFG as ambiguous: the same sentence can be obtained by several derivation trees. This implies that there are several individuals representing the solution to a problem. It is therefore easier to find. This operator consists of eight steps:

1. Choose a node, except the axiom, with a non-terminal symbol randomly from the first parent. This node is called crossover node and is denoted CN1.
2. Choose the parent of CN1. As we are working with a CFG, this will be a non-terminal symbol. The right-hand sides of all its production rules are stored in the array R.
3. The derivation produced by the parent of CN1 is called main derivation, and is denoted A ::= C. Calculate the derivation length l as the number of symbols in the right-hand side of the main derivation. Having l, the position (p) of CN1 in the main derivation and C, define the three-tuple T(l, p, C).
4. Delete from R all the right-hand sides with different lengths from the main derivation.
5. Remove from R all those right-hand sides in which there exists any difference between the symbols (except the one located in position p) in each right-hand side and the symbols in C.
6. The set X is formed by all the symbols in the right-hand sides of R that are in position p. X contains all the non-terminal symbols of the second parent that can be chosen as a crossover node (CN2).
7. Choose CN2 randomly from X, discarding all the nodes that will generate offspring trees with a size greater than a previously established value D.

8. Calculate the two new derivation trees produced as offspring by swapping the two subtrees whose roots are CN1 and CN2.

The underlying idea of this algorithm consists on calculating which are the non-terminal symbols that can substitute the symbol contained in CN1, bearing in mind that the production rule that contains CN1 keeps being valid. Since all non-terminal symbols that can generate valid production rules are taken into account in the crossover process, this operator takes advantage of ambiguous grammars.

The proposed crossover operator has primarily three attractive features: a) step 7 states a code bloat control mechanism, b) the offspring produced are always composed of two valid trees and c) step 6 indicates that all the possible nodes of the second parent that can generate valid individuals are taken into account, not only those nodes with the same non-terminal symbol as the one chosen for the first parent. This third feature increases the GGGP system's exploration capability, which avoids trapping in local optima and takes advantage of there being more than one derivation tree (potential solution to the problem) for a single sentence.

Results

We present and discuss the results achieved by the crossover operators described in the background section and the operator that we propose. To do so, we have tackled a complex classification problem: the real-world task of providing breast cancer prognosis (benign or malignant) from the morphological characteristics of microcalcifications. Microcalcifications are small mineral deposits in breast tissue that could constitute cancer. This experiment involved searching a knowledge base of fuzzy rules that could give such a prognosis.

The data employed for giving a disease prognosis are: patient's age, lesion size, lesion location in the breast, and particular features of the microcalcifications: number, distribution and type. Number indicates the quantity of existing clustered microcalifications, distribution shows how they are clustered and type reflects the individual morphology of the microcalcifications. To run the tests, 365 microcalcifications were selected at random. Of these, 315 lesions were randomly selected for use as genetic programming system training cases with the different crossover operators described. After training, the fittest individual was selected to form a knowledge base with the fuzzy rules encoded by this individual. Then, the knowledge base was tested with

Figure 3. Average convergence speed for each crossover operator
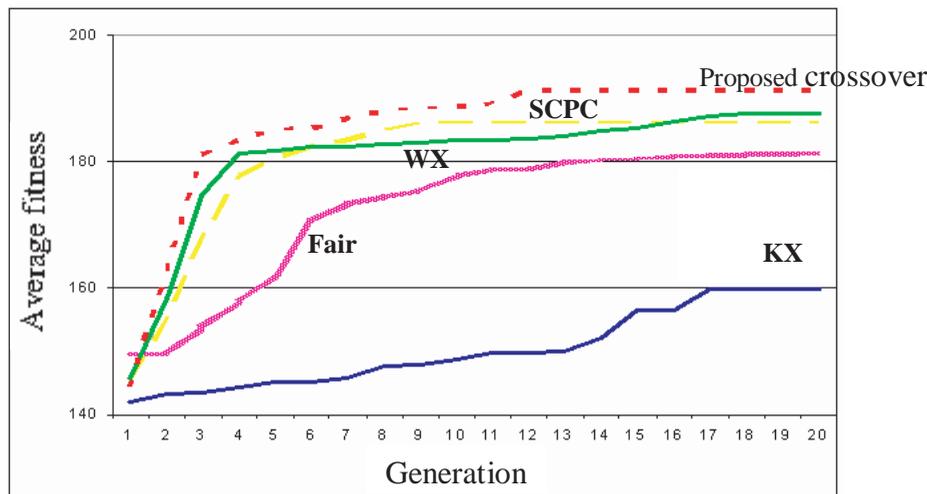
the 50 remaining lesions not chosen during the training phase to output the number of correctly classified patterns in what we have called the testing phase.

The CFG employed was formed by 19 non-terminal symbols, 54 terminals and 51 production rules, some of them included to obtain an ambiguous grammar. The population size employed was 1000, the upper bound for the size of the derivation trees was set to 20. The fitnesss function consisted of calculating the number of well-classified patterns. Therefore, the greater the fitness, the fitter the individual is, with the maximum limit of 315 in the training phase and 50 in the test.

Figure 3 shows the average evolution process for each of the five crossover operators in the training phase after 100 executions.

It is clear from Figure 3 that KX yields the worst results, because it maintains an over-diverse population and allows invalid individuals to be generated. This prevents it from focusing on one possible solution. The effect of Fair is just the opposite, leading very quickly to one of the optimal solutions (this is why it has a relatively high convergence speed initially), and

slowing down if convergence is towards a local optimum (which happens in most cases). WX and SCPC produce good results, bettered only by the proposed crossover. Its high convergence speed evidences the benefits of taking into account all possible nodes of the second parent that can generate valid offspring.

Table 1 shows examples of fuzzy rules output in one of the executions for the best two crossover operators —WX and the proposed operator— once the training phase was complete.

Table 2 shows the average number (rounded up or down to the nearest integer) of correctly classified patterns after 100 executions, achieved by the best individual in the training and test phases, and the percentage of times that the system converged prematurely.

KX again yields the worst results, correctly classifying just 57.46% (181/315) of patterns in the training phase and 54% (27/50) in the testing phase. SCPC and Fair crossovers also return insufficient results: around 59% in the training phase and 54%-56% in the testing phase, although, as shown in Figure 3, SCPC has a higher convergence speed. Finally, note the similarity

Table 1. Some knowledge base fuzzy rules output by two GGGP systems

Crossover operator	Rule 1	Rule 2
WX	IF NOT (type=branched) OR (number=few) THEN (prognosis=benign)	
Proposed	IF NOT (age=middle) AND NOT (location=subaerolar) AND NOT(type=oval) THEN (prognosis=malignant)	IF (type=heterogeneous) THEN (prognosis=malignant)

Table 2. Average number of correctly classified patterns and unsuccessful runs

Crossover operator	Training	Testing	Unsuccessful runs
KX	181/315 (57.46%)	27/50 (54%)	36%
SCPC	186/315 (59.04%)	28/50 (56%)	14%
Fair	185/315 (58.73%)	27/50 (54%)	15%
WX	191/315(60.63%)	30/50 (60%)	8%
Proposed	191/315(60.63%)	31/50 (62%)	2%

between WX and the proposed operator. However, the proposed operator has higher speed of convergence and is less likely to get trapped in local optima, as it converged prematurely only twice in 100 executions.

FUTURE TRENDS

The continuation of the work described in this article can be divided into two main lines of investigation in GGGP. The first involves finding an algorithm that can estimate the maximum sizes of the trees generated throughout the evolution process to assure that the optimal solution will be reached. This would overcome the proposed crossover operator's weakness of not being able to reach a solution because the permitted maximum tree size is too restrictive for it to be able to reach a good solution, whereas this solution could be found if individuals were just a little larger.

The second interesting line of research derived from this work is the use of ambiguous grammars. It has been empirically observed that using the proposed operator combined with ambiguous grammars in GGGP systems benefits convergence speed. However, "too much ambiguity" is damaging. The idea is to get an ambiguity measure that can answer the question of how much ambiguity is needed to get the best results in terms of efficiency.

CONCLUSION

This article summarizes the latest and most important advances in GGGP, paying special attention to the crossover operator, which (alongside the initialization method, the codification of individuals and, to a lesser extent, the mutation operator, of course) is chiefly responsible for the convergence speed and the success of the evolution process.

GGGP systems are able to find solutions to any problem that can be syntactically expressed by a CFG. The proposed crossover operator provides GGGP systems with a satisfactory balance between exploration and exploitation capabilities. This results in a high convergence speed, while eluding local optima as the reported results demonstrate. To be able to achieve such good results, the proposed crossover operator includes a computationally cheap mechanism to control bloat, it always generates syntactically valid offspring and it

can choose any node from the second parent to generate the offspring, rather than just those nodes with the same non-terminal symbols as the one chosen in the first parent.

REFERENCES

Barrios, D., Carrascal, A., Manrique, D. & Ríos, J. (2003). Optimization with real-coded genetic algorithms based on mathematical morphology. *International Journal of Computer Mathematics*, (80) 3, 275-293.

Couchet, J., Manrique, D., Ríos, J. & Rodríguez-Patón, A. (2006). Crossover and mutation operators for grammar-guided genetic programming. *Softcomputing*, DOI 10.1007/s00500-006-0144-9.

Crawford-Marks, R. & Spector, L. (2002). Size control via size fair genetic operators in the pushGP genetic programming system. *In proceedings of the genetic and evolutionary computation conference*, New York, 733-739.

D'haesler, P. (1994). Context preserving crossover in genetic programming. *In IEEE Proceedings of the 1994 world congress on computational intelligence*, Orlando, (1) 379-407

Dounias, G., Tsakonas, A., Jantzen, J., Axer, H., Bjerregard, B., & von Keyserlingk, D. (2002). Genetic Programming for the Generation of Crisp and Fuzzy Rule Bases in Classification and Diagnosis of Medical Data. *Proceedings of the 1st International NAISO Congress on Neuro Fuzzy Technologies,* Havana, Cuba, 494-500.

Grosman, B. & Lewin, D.R. (2004). Adaptive Genetic Programming for Steady-State Process Modeling. *Computers and Chemical Engineering*, 28 2779-2790.

Hussain, T.S. (2003). *Attribute grammar encoding of the structure and behaviour of artificial neural networks*. PhD Thesis, Queen's University. Kingston, Ontario, Canada.

Keedwell, E., & Narayanan, A. (2005). *Intelligent bioinformatics*. Wiley & Sons.

Koza, JR. (1992). *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge.

G

Langdon, WB. (1999). Size fair and homologous tree genetic programming crossovers. *In proceedings of genetic and evolutionary computation conference*, GECCO'99, Washington DC, 1092-1097.

Manrique, D. (2001). *Diseño de redes de neuronas y nuevas técnicas de optimización mediante algoritmos genéticos [Artificial neural networks design and new optimization techniques using genetic algorithms]*. PhD Thesis, Facultad de Informática, Universidad Politécnica de Madrid.

Manrique, D., Márquez, F., Ríos, J. & Rodríguez-Patón A. (2005). Grammar-based crossover operator in genetic programming. *Lecture Notes in Artificial Intelligence*, 3562 252-261.

Rodrigues, E. & Pozo, A. (2002). Grammar-Guided Genetic Programming and Automatically Defined Functions. *In proceedings of the 16th Brazilian symposium on artificial intelligence*, Recife, Brazil, 324-333.

Terrio, MD., & Heywood, MI. (2002). Directing crossover for reduction of bloat in GP. *In IEEE proceedings of Canadian conference on electrical and computer engineering*, (2) 1111-1115.

Whigham, P.A. (1995). Grammatically-based genetic programming. *In proceedings of the workshop on genetic programming: from theory to real-world applications*, California, 33-41.

Whigham, P.A. (1996). *Grammatical bias for evolutionary learning*. PhD Thesis, School of Computer Science, Australian Defence Force (ADFA), University College, University of New South Wales.

Yao, X., & Xu, Y. (2006). Recent advances in evolutionary computation. *Journal of Computer Science & Technology*, (21) 1 1-18.

KEY TERMS

Ambiguous Grammar: Any grammar in which different derivation trees can generate the same sentence.

Closure Problem: Phenomenon that involves always generating syntactically valid individuals.

Code Bloat: Phenomenon to be avoided in a genetic programming system convergence process involving the uncontrolled growth, in terms of size and complexity, of individuals in the population

Convergence: Process by means of which an algorithm (in this case an evolutionary system) gradually approaches a solution. A genetic programming system is said to have converged when most of the individuals in the population are equal or when the system cannot evolve any further.

Fitness: Measure associated with individuals in an evolutionary algorithm population to determine how good the solution they represent is for the problem.

Genetic Programming: A variant of genetic algorithms that uses simulated evolution to discover functional programs to solve a task.

Grammar-Guided Genetic Programming: The application of analytical methods and tools to data for the purpose of identifying patterns, relationships or obtaining systems that perform useful tasks such as classification, prediction, estimation, or affinity grouping.

Intron: Segment of code within an individual (subtree) that does not modify the fitness, but is on the side of convergence process.

Granular Computing

Georg Peters
Munich University of Applied Sciences, Germany

INTRODUCTION

It is well accepted that in many real life situations information is not certain and precise but rather uncertain or imprecise. To describe uncertainty probability theory emerged in the 17th and 18th century. Bernoulli, Laplace and Pascal are considered to be the fathers of probability theory. Today probability can still be considered as the prevalent theory to describe uncertainty.

However, in the year 1965 Zadeh seemed to have challenged probability theory by introducing fuzzy sets as a theory dealing with uncertainty (Zadeh, 1965). Since then it has been discussed whether probability and fuzzy set theory are complementary or rather competitive (Zadeh, 1995). Sometimes fuzzy sets theory is even considered as a subset of probability theory and therefore dispensable. Although the discussion on the relationship of probability and fuzziness seems to have lost the intensity of its early years it is still continuing today. However, fuzzy set theory has established itself as a central approach to tackle uncertainty. For a discussion on the relationship of probability and fuzziness the reader is referred to e.g. Dubois, Prade (1993), Ross et al. (2002) or Zadeh (1995).

In the meantime further ideas how to deal with uncertainty have been suggested. For example, Pawlak introduced rough sets in the beginning of the eighties of the last century (Pawlak, 1982), a theory that has risen increasing attentions in the last years. For a comparison of probability, fuzzy sets and rough sets the reader is referred to Lin (2002).

Presently research is conducted to develop a Generalized Theory of Uncertainty (GTU) as a framework for any kind of uncertainty whether it is based on probability, fuzziness besides others (Zadeh, 2005). Cornerstones in this theory are the concepts of information granularity (Zadeh, 1979) and generalized constraints (Zadeh, 1986).

In this context the term Granular Computing was first suggested by Lin (1998a, 1998b), however it still lacks of a unique and well accepted definition. So, for example, Zadeh (2006a) colorfully calls granular computing "ballpark computing" or more precisely "a mode of computation in which the objects of computation are generalized constraints".

BACKGROUND

Humans often speak and think in words rather than in numbers. For example, in summer we say that it is *hot* outside rather than that is *35.32° Celsius*. This means that we often define our information as an imprecise *perception-based linguistic variable* rather than as a precise *measure-based number*. The impreciseness in our formulation basically has four reasons (Zadeh, 2005):

1. *Bounded ability of human sensors and computational limits of the brain*. (1) Our human sensors do not have the abilities of a laser based speed controller. So we cannot quantify the speed of a racing car as *252.18 km/h* in Albert Park, Melbourne. However on the linguistic level we can define the car as *fast*. (2) M∈ost people cannot numerically calculate the exact race distance given by *5,303 km * 53 turns=307.574 km* due to computational limits of their brains. However they probably estimate that it will be *around 300 km*.
2. *Lack of numerical information*. Melbourne is considered as a shopping paradise in Australia since there are *countless* shops. Maybe only local government knows the exact number of shops.
3. *Qualitative, non quantifiable information*. Much information is provided rather qualitative than quantitative. If one describes the quality of a pizza in an Italian restaurant in Lygon Street in Melbourne's suburb Carlton only a qualitative, linguistic judgment like *excellent* or *very good* is possible. The judgment is hardly to be quantifiable (beside a technical counting of the olives or the weight of the salami etc.).

4. *Tolerance for imprecision.* Recall the example, Melbourne as a shopping paradise, given above. To define Melbourne as shopping paradise its exact number of shops is not needed. It is sufficient to know that there are many shops. This tolerance for impression often makes a statement more robust and efficient in comparison to exact numerical values.

So obviously humans often prefer not to deal with precise but favor vague information that is immanent in natural language.

Humans would rarely formulate a sentence like:

With a probability of 97.34% I will see Ken, who has a height of 1.97m, at 12:05pm.

Instead most humans would prefer to say:

Around noon I will almost certainly meet tall Ken.

While the first formulation is computer compatible since it contains numbers (singletons) the second formulation seems too be to imprecise to be used as input for computers.

A central objective of the concept of granular computing is to bridge this gap and compute with words (Zadeh, 1996). This leads to the ideas of information granularity or granular computing which was introduced by Zadeh (1986, 1979).

The concept of information granularity has its roots in fuzzy set theory (Zadeh, 1965, 1997). Zadeh (1986) advanced and generalized this idea so that granular computing subsumes any kind of uncertainty and imprecision like "set theory and interval analysis, fuzzy sets, rough sets, shadowed sets, probabilistic sets and probability [...], high level granular constructs" (Bargiela, Pedrycz, 2002, p. 5). The term granular computing was first suggested by Lin (1998a, 1998b).

FUNDAMENTALS OF GRANULAR COMPUTING

Singular and Granular Values

To more formally describe the difference between natural language and precise information let us recall the example sentences given in Section 2. The infor-

Figure 1. Mapping of Singletons and granular values
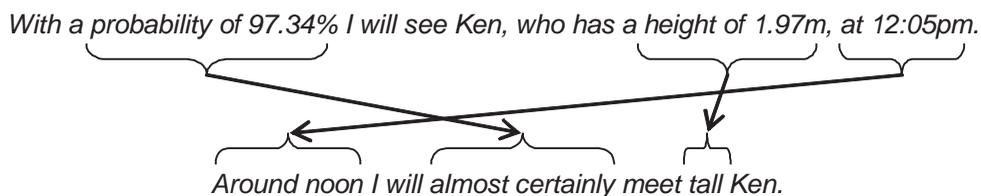
Table 1. Singular and granular values

Variable	Singular Values	Granular Values
Probability	97.34%	almost certainly
Height	1.97m	tall
Time	12:05pm	around noon

mation given in the two sentences can be mapped as depicted in Figure 1.

While the first sentence contains exact figures (singletons) the second sentence describes the same context using linguistic variables (granular values). A comparison of the singular and granular values is given in Table 1.

For example, the variable height can be mapped to the singleton *1.97m* or the granule *tall*. The granule *tall* covers not only the singleton *1.97m* but also neighbor-

hood values. See Figure 2 for an interval granulation of the singleton of the variable height; a fuzzy membership function (linguistic variable) would be another possibility for a granule of *tall* (see Figure 3).

The main difference in the representation of the variable heights is entailed by a different formulation of the constraints. While the formulation as a singleton is of bivalence nature (height=1.97m) a fuzzy formulation would contain memberships. This leads to the concept of generalized constraints.

Figure 2. Presentation of variable height as Singleton and granule
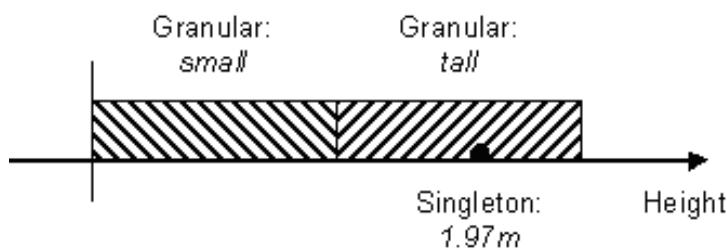
Figure 3. Fuzzy memberships
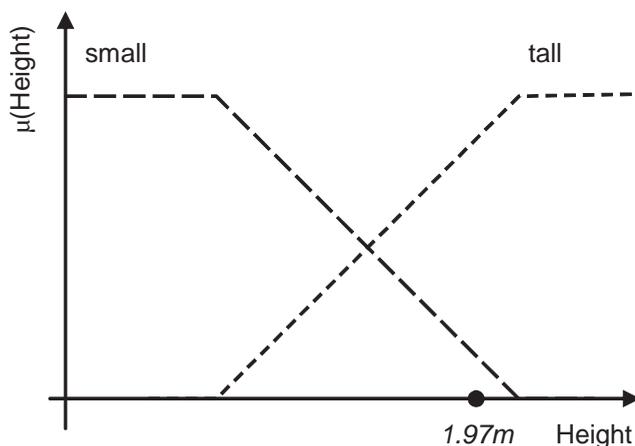
Generalized Constraints

Overview of Constraints

The generalization of constraints is a central concept in granular computing. The main purpose is to make classic constraints like ∈ (member), = (equal); < (smaller) and > (greater) more flexible and therefore closer to the way humans think. In the following subsections we will discuss standard, primary and general constraints in more detail.

Basic Concept of Generalized Constraints

Standard Constraints. A *standard constraint* C is characterized by its bivalency (possibilistic of veristic) or probabilistic nature. Bivalent and probabilistic constraints do not have memberships degrees which indicate the degree of satisfaction of the constraint A: a variable X does or does not fulfill the standard constraint. Examples for bivalent constraints are: ∈ (member), = (equal); < (smaller) and > (greater) besides others.

Primary Constraints. Zadeh (2006a) suggested the following *primary constraints*:

- Possibilistic (r=blank)
- Probabilistic (r=p)
- Veristic (r=v)

since they formulate the basic perceptions possibility, likelihood and truth. In contrast to the standard constraints bivalency is no longer required for the possibilistic and veristic constraints. Therefore standard constraints are included in the primary constraints.

Applying the primary constraints to our example the second "Ken sentence" of Section 2 we get:

- Possibilistic Constraint (X is R): *Ken is tall → Height(Ken) is tall* (see Dubois, Prade (1998) for semantics of fuzzy sets including possibility (Zadeh, 1978)).
- Probabilistic Constraint (X isp R): *Actual arrival time (X) at meeting point → X isp N(μ, σ²)* is e.g. normal distributed around the agreed meeting time μ.
- Veristic Constraint (X isv R): *Ken is at the meeting point at 12:05pm → Present(Ken, meeting point) isv 12:05pm.*

Generalized Constraints. Further constraints include (Zadeh, 2005) usuality (r=u), random set (r=rs), fuzzy graph (r=fg), bimodal (r=bm) and group (r=g). The set of general constraints consists of these and the primary constraints. So, formally a generalized constraints (GC) is given by (Zadeh, 2005):

GC(X): X isr R

with X the constrained variable and R the non-bivalent relation. In the term *isr* the letter *r* defines the semantics or the modality of the constraint as describe above.

Generalized Constraint Language

To formally describe generalized constraints Zadeh (2006b) suggests a Generalized Constraint Language (GCL). In Section 3.2.2 we already used the GCL in the presented example, e.g. the mapping: *Ken is tall → Height(Ken) is tall*, which has the form

$p → X\ isr\ R$

with p an expression in natural language. In this context Zadeh (2006b) defines the translation of natural language into GCL as *precisiation*. The precisiation can lead to *v-precise* and/or *m-precise* results:

- v-precisiation: a precise value is obtained. v-precisiation has *s-precisiation* (singleton), *cg-precisiation* (crisp granular) and *g-precisiation* (granular) as its modalities. s-precisiation leads to a singleton, while cg-precisiation leads to an crisp interval. g-precisiation is the most general form of precisiation and leads to fuzzy intervals, fuzzy graphs besides others.
- m-precisiation: a precise meaning is obtained. m-precisiation can further divided into the modalities mm-precisiation (machine-oriented) and mh- precisiation (human-oriented).

Examples: (1) *Ken is between a and b meters tall* is m-precise and since the variables *a* and *b* are not specified v-imprecise. (2) *Ken is approximately c meters tall → Ken is a meters tall* is a s-precisiation. The term *approximately c* can also be abbreviated as *c**. The star indicates that *c* is a granular value.

Figure 4. Rough sets
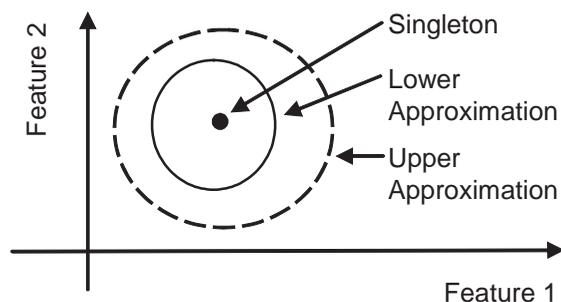
In contrast to precisiation granulation leads to an imprecisiation of the information. Obviously the translation *Ken is 1.97m → Ken is c meters tall* is a v-imprecisiation and *Ken is c meters tall → Ken is tall* a m-imprecisiation.

So for example, rough sets can be interpreted as cascading cg-imprecisiation. In rough set theory (Pawlak, 1982) a set is described by a lower and upper approximation ($_LA$ and ^UA respectively). The lower approximation is a subset of the upper approximation. While the objects in the lower approximation surely belong to the corresponding set the objects in a upper approximation might belong to the set.

Therefore rough set theory provides an example of a cascading granulation: $X Î _LA Ì {}^UA$ (see Figure 4).

Deduction Rules

Principal Deduction Rules

In this Section we regard the term granular computing in its literally meaning: how to compute with granules and focus on principal deductions (Zadeh, 2005, 2006b):

- Conjunction
- Projection
- Protagation

For more details on deduction rules the reader is referred to Zadeh (2005, 2006b).

Generalized Extension Principle

One of the most fundamental theorem in fuzzy logic is the Extension Principle (Zadeh, 1975, Zimmermann, 2001). Basically the Extension Principle defines how the memberships $\mu_y(y)$ of an endogenous variable

Y=f(X)

can be determined with *X* and *Y* singletons and $\mu_x(X)$ given. A simple transformation $\mu_y(Y)= \mu_y(f(X))= \mu_x(X)$ does not generally provide a unique solution. Therefore, to obtain a unique solution, *sup $\mu_y(f(X))$* is taken.

The Generalized Extension Principle (Zadeh, 2006a) establishes a relationship between

Y*=f*(X*)

Gr(Y) isr Gr(X)

with *Y**, *X** and *f*()* granules. It can be considered as primary deduction rule since many others deduction rules can be derived from it (Zadeh, 2006b).

Example

Let us consider an example (Zadeh, 2005, 2006a, 2006b):

The following linguistic statement is given:

Most Swedes are tall → (Height(Swedes) are tall) is most.

First let us specify

Swedes are tall → ∫ X(h)μ_{tall}(h)dh

with *X(h)* the height density function and $\mu_{tall}(h)$ the membership function for the linguistic variable *tall*.

Second we have to apply the linguistic variable *most* to the expression *Swedes are tall* and obtain:

Most (Swedes are tall) → μ_{mostl} (∫ X(h)μ_{tall}(h)dh)

As result we get a precise formulation of the given linguistic statement.

CONCLUSION AND FUTURE RESEARCH

Granular Computing is a mighty framework to deal with uncertainty. Information granules can include probabilistic as well as possibilistic phenomena besides others. Therefore granular computing functions as a umbrella for them without competing with them. One core advantage is that is helps to bridge the gap between (imprecise) natural language and the precision that is immanent in computers etc. Presently Zadeh is promoting his idea towards a Generalized Theory of Uncertainty in many publications and presentations. In future the Generalized Theory of Uncertainty will probably be the dominant label for anything related to this topic. Since the Generalized Theory of Uncertainty is a young but rapidly emerging new branch in science future research will go in the direction of the generalization of uncertainty concepts, e.g. from probabilistic and fuzzy clustering towards granular clustering.

REFERENCES

Bargiela, A. & Pedrycz, W. (2002). *Granular computing: an introduction*. Boston: Kluwer Acamemic Publishers.

Dubois, D. and Prade, H. (1993). Fuzzy sets and probability: misunderstandings, bridges and gaps.. In *Proceedings of the second IEEE International Conference on Fuzzy Systems* (pp. 1059-1068), San Francisco.

Dubois, D. and Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, 90, 141-150.

Lin, T.Y. (1998a). Granular Computing on Binary Relations I: Data Mining and Neighborhood Systems. In Skowron, A. and Polkowski, L. (eds.), *Rough Sets In Knowledge Discovery* (pp. 107-121), Heidelberg: Physica-Verlag.

Lin, T.Y. (1998a). Granular Computing on Binary Relations II: Data Mining and Neighborhood Systems. In Skowron, A. and Polkowski, L. (eds.), *Rough Sets In Knowledge Discovery* (pp. 121-140), Heidelberg: Physica-Verlag.

Lin, T.Y. (2002). Fuzzy sets, rough set and probability. In Keller, J. and Nasraoui, O. (eds), *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society 2002* (pp.302-305), University, New Orleans.

Pawlak, Z. (1982). Rough sets. *International Journal of Parallel Programming*, 11, 341-356.

Ross, T.J.; Booker. J.M.; Parkinson, W.J. (2002). *Fuzzy Logic and Probability Applications: A Practical Guide*. Philadelphia: SIAM - Society for Industrial & Applied Mathematics.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8, 338-353.

Zadeh, L. (1978). Fuzzy sets as the basis for a theory of possibility. *Fuzzy Sets and Systems*, 1, 3-28.

Zadeh, L. (1979). Fuzzy Sets and Information Granularity. In Gupta, M., Ragade, R., Yager, R. (eds.), *Advances in Fuzzy Set Theory and Applications* (pp. 3-18). Amsterdam: North-Holland Publishing.

Zadeh, L. (1986). Outline of a computational approach to meaning and knowledge representation based on the concept of a generalized assignment statement. In Thoma, M. and Wyner A. (eds.), *Proceedings of the International Seminar on Artificial Intelligence and Man-Machine Systems* (LNCIS 80, pp. 198-211). Heidelberg: Springer-Verlag.

Zadeh, L. (1995). Discussion: probability theory and fuzzy logic are complementary rather than competitive. *Technometrics*, 37, 271-276.

Zadeh, L. (1996). Fuzzy logic = computing with words. *IEEE Transactions of Fuzzy Systems*, 2, 103-111.

Zadeh, L. (1997). Towards a theory of fuzzy information granularity and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90, 111-127.

Zadek, L. (2005). *Toward a generalized theory of uncertainty (GTU)—an outline*. Information Sciences, 172, 1-40.

Zadeh, L. (2006a). Granular computing - the concept of generalized-constraint-based computing (presentation slides). *Proceedings of Rough Sets and Current Trends in Computing: 5th International Conference,* (LNCS 4259, pp 12-14). Heidelberg: Springer-Verlag.

Zadeh, L. (2006b). *Generalized theory of uncertainty (GTU) - principle concepts and ideas*. Computational Statistics & Data Analysis, 51, 15-46

G

Zimmermann, H. J. (2001). *Fuzzy set theory and its applications*. Boston: Kluwer Academic Publishers.

KEY TERMS

Fuzzy Set Theory: Fuzzy set theory was introduced by Zahed in 1965. The central idea of fuzzy set theory is that an object belongs to more than one sets simultaneously. the closeness of the object to a set is indicated by membership degrees.

Generalized Theory of Uncertainty (GTU): GTU is a framework that shall subsume any kind of uncertainty (Zadeh 2006a). The core idea is to formulate generalized constraints (like possibilistic, probabilistic, veristic etc.). The objective of GTU is not to replace existing theories like probability or fuzzy sets but to provide an umbrella that allows to formulate any kind of uncertainty in a unique way.

Granular Computing: The idea of granular computing goes back to Zadeh (1979). The basic idea of granular computing is that an object is describe by a bunch of values in possible dimensions like indistinguishability, similarity and proximity. If a granular is labeled by a linguistic expressing it is called a linguistic variable. Zahed (2006a) defines granular computing as "a mode of computation in which the objects of computation are generalized constraints".

Hybridization: Combination of methods like probabilistic, fuzzy, rough concepts, or neural nets, e.g. fuzzy-rough, rough-fuzzy or probabilistic-rough, or fuzzy-neural approaches.

Linguistic Variable: A linguistic variable is a linguistic expression (one or more words) labeling an information granular. For example a membership function is labeled by the expressions like "hot temperature" or "rich customer".

Membership Function: A membership function shows the membership degrees of a variable to a certain set. For example, a temperature t=30° C belongs to the set "hot temperature" with a membership degree $\lambda_{HT}(30°)=0.8$. The membership functions are not objective but context and subject-dependent.

Rough Set Theory: Rough set theory was introduced by Pawlak in 1982. The central idea of rough sets is that some objects distinguishable while others are indiscernible from each other.

Soft Computing: In contrast to "hard computing" soft computing is collection of methods (fuzzy sets, rough sets neutral nets etc.) for dealing with ambiguous situations like imprecision, uncertainty, e.g. human expressions like "high profit at reasonable risks". The objective of applying soft computing is to obtain robust solutions at reasonable costs.

Growing Self–Organizing Maps for Data Analysis

Soledad Delgado
Technical University of Madrid, Spain

Consuelo Gonzalo
Technical University of Madrid, Spain

Estíbaliz Martínez
Technical University of Madrid, Spain

Águeda Arquero
Technical University of Madrid, Spain

INTRODUCTION

Currently, there exist many research areas that produce large multivariable datasets that are difficult to visualize in order to extract useful information. Kohonen self-organizing maps have been used successfully in the visualization and analysis of multidimensional data. In this work, a projection technique that compresses multidimensional datasets into two dimensional space using growing self-organizing maps is described. With this embedding scheme, traditional Kohonen visualization methods have been implemented using growing cell structures networks. New graphical map displays have been compared with Kohonen graphs using two groups of simulated data and one group of real multidimensional data selected from a satellite scene.

BACKGROUND

Data mining first stage usually consist of building simplified global overviews of data sets, generally in graphical form (Tukey, 1977). At present, the huge amount of information and its multidimensional nature complicates the possibility to employ direct graphic representation techniques. Self-Organizing Maps (Kohonen, 1982) fit well in the exploratory data analysis since its principal purpose is the visualization and the analysis of nonlinear relations between multidimensional data (Rossi, 2006). In this sense, a great variety of Kohonen's SOM visualization techniques (Kohonen, 2001) (Ultsch & Siemon, 1990) (Kraaijveld,

Mao & Jain, 1995) (Merlk & Rauber, 1997) (Rubio & Giménez 2003) (Vesanto, 1999), and some automatic map analysis (Franzmeier, Witkowski & Rückert 2005) have been proposed.

In Kohonen's SOM the network structure has to be specified in advance and remains static during the training process. The choice of an inappropriate network structure can degrade the performance of the network. Some growing self-organizing maps have been implemented to avoid this disadvantage. In (Fritzke, 1994), Fritzke proposed the Growing Cell Structures (GCS) model, with a fixed dimensionality associated to the output map. In (Fritzke, 1995), the Growing Neural Gas is exposed, a new SOM model that learns topology relations. Even though the GNG networks get best grade of topology preservation than GCS networks, due to the multidimensional nature of the output map it cannot be used to generate graphical map displays in the plane. However, using the GCS model it is possible to create networks with a fixed dimensionality lower or equal than 3 that can be projected in a plane (Fritzke, 1994). GCS model, without removal of cells, has been used to compress biomedical multidimensional data sets to be displayed as two-dimensional colour images (Walker, Cross & Harrison, 1999).

GROWING CELL STRUCTURES VISUALIZATION

This work studies the GCS networks to obtain an embedding method to project the bi-dimensional output

map, with the aim of generating several graphic map displays for the exploratory data analysis during and after the self-organization process.

Growing Cell Structures

The visualization methods presented in this work are based on self-organizing map architecture and learning process of Fritzke's Growing Cell Structures (GCS) network (Fritzke, 1994). GCS network architecture consists of connected units forming k-dimensional hypertetrahedron structures linked between them. The interconnection scheme defines the neighbourhood relationships. During the learning process, new units are added and superfluous ones are removed, but these modifications are performed in such way that the original architecture structure is maintained.

The training algorithm is an iterative process that performs a non-linear projection of the input data over the output map, trying to preserve the topology of the original data distribution. The self-organization process of the GCS networks is similar that in Kohonen's model. For each input signal the best matching unit (*bmu*) is determined, and *bmu* and its direct neighbour's synaptic vectors are modified. In GCS networks each neuron has associated a resource, which can represent the number of input signals received by the neuron, or the summed quantization error caused by the neuron. In every adaptation step the resource of the *bmu* is conveniently modified. A new neuron is inserted between the unit with highest resource, q, and its direct neighbour with the most different reference vector, f, after a fixed number of adaptation steps. The new unit synaptic vector is interpolated from the synaptic vectors of q and f, and the resources values of q and f are redistributed too. In addition, neighbouring connections are modified in order to ensure the output architecture structure. Once all the training vectors have been processed a fixed number of times (epoch), the neurons whose reference vectors fall into regions with a very low probability density are removed. To guarantee the architecture structure some neighbouring connections are modified too. Relative normalized probability density estimation value proposed in (Delgado, 2004) has been used in this work to determine the units to be removed. This value provides better interpretation of some training parameters, improving the removal of cells and the topology preserving of the network.

Several separated meshes could appear in the output map when superfluous units are removed.

When the growing self-organization process finishes, the synaptic vectors of the output units along with the neighbouring connections can be used to analyze different input space properties visually.

Network Visualization: Constructing the Topographic Map

The ability to project high-dimensional input data onto a low-dimensional grid is an important property of Kohonen feature maps. By drawing the output map over a plane it will be possible to visualize complex data and discover properties or relations of the input vector space not expected in advance. Output layer of Kohonen feature maps can be printed on a plane easily, painting a rectangular grid, where each cell represents an output neuron and neighbour cells correspond to neighbour output units.

GCS networks have less regular output unit connections than Kohonen ones. When $k=2$ architecture factor is used, the GCS output layer is organized in groups of interconnected triangles. In spite of bi-dimensional nature of these meshes, it is not obvious how to embed this structure into the plane in order to visualize it. In (Fritzke, 1994), Fritzke proposed a physical model to construct the bi-dimensional embedding during the self-organization process of the GCS network. Each output neuron is modelled by a disc, with diameter d, made of elastic material. Two discs with distance d between centres touch each other, and two discs with distance smaller than d repeal each other. Each neighbourhood connection is modelled as an elastic string. Two discs connected but not touching are pulled each other. Finally, all discs are positively charged and repeal each other. Using this model, the bi-dimensional topographic coordinates of each output neuron can be obtained, and thus, the bi-dimensional output meshes can be printed on a plane.

In order to obtain the output units bi-dimensional coordinates of the topographic map (for $k=2$), a slightly modified version of this physical model has been used in this contribution. At the beginning of the training process, the initial three output neurons are placed in the plane in a triangle form. Each time a new neuron is inserted, its position in the plane is located exactly halfway of the position of the two neighbouring neurons between which it has been inserted. After this, attraction

and repulsion forces are calculated for every output neuron and its positions are consequently moved. The attraction force of a unit is calculated as the sum of individual attraction forces that all neighbouring connections exercise over it. Attraction force between two neighbouring neurons i and j, with p_i and p_j coordinates in the plane, and Euclidean distance e, is calculated as $(e-d)/2$ if $e \geq d$, and 0 otherwise. The repelling force of a unit is calculated as the sum of individual repulsion forces that all no-neighbouring output neurons exercise over it. Repelling force between two no-neighbouring neurons i and j is calculated as $d/5$ if $2d<e\leq 3d$, $d/2$ if $d<e\leq 2d$, d if $0<e\leq d$, and 0 otherwise. There exist three basic differences between the embedding model used in this work and the Fritzke's one. First, repelling force is only calculated with no-neighbouring units. Second, attracting force between two neurons i and j is multiplied by the distance normalization $((p_j-p_i)/e)$ and by the attraction factor 0.1 (instead of 1). Last, repelling force between two neurons i and j is multiplied by the distance normalization $((p_i-p_j)/e)$ and by the repulsion factor 0.05 (instead of 0.2).

The result of applying this projection method is showed in Fig. 1. When removal of cells is performed, different meshes are showed unconnectedly. Without any other additional information, this projection method makes possible cluster detection.

Visualization Methods

Using the projection method exposed, traditional Kohonen visualization methods can be implemented using GCS networks with $k=2$. Each output neuron is painted as a circle in a colour determined by a major parameter. When greyscale is used, normally dark and clear tones are associated with high and low values respectively. The grey scales are relative to the maximum and minimum values taken by the parameter. The nature of the data used to calculate the parameter determines three general types of methods for performing visual analysis of self-organizing maps: distances between synaptic vectors, training patterns projection over the neurons, and individual information about synaptic vectors.

All the experiments have been performed using two groups of simulated data and one group of real multidimensional data (Fig. 2) selected from a scene registered by the ETM+ sensor (Landsat 7). The input signals are defined by the six ETM+ spectral bands with

Figure 1. Output mesh projection during different self-organization process stages of a GCS network trained with bi-dimensional vectors distributed on eleven separate regions.

(a) (b) (c) (d) (e)
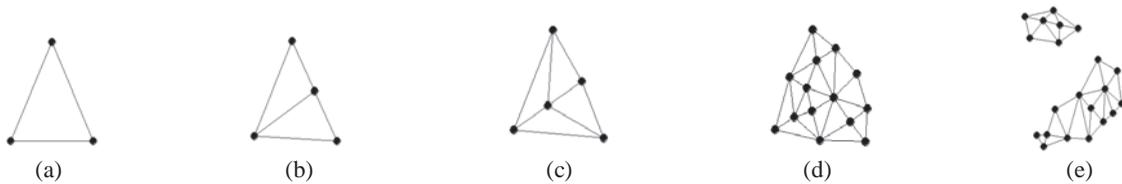
Figure 2. (a) Eleven separate regions in the bi-dimensional plane. (b) Two three dimensional chain-link. (c) Projection of multidimensional data of satellite image.

(a) (b) (c)
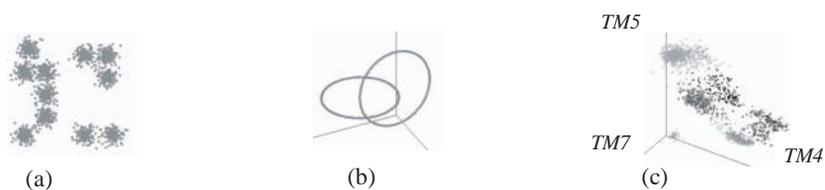
the same spatial resolution: TM1 to TM5, and TM7. The input data set has a total number of 1800 pixels, 1500 carefully chosen from the original scene and 300 randomly selected. The input vectors are associated to six land cover categories.

Displaying Distances

The adaptation process of GCS networks places the synaptic vectors in regions with high probability density, removing units positioned into regions with a very low probability density. A graphical representation of distances between the synaptic vectors will be a useful tool to detect clusters over the input space. Distance map, unified distance map (U-map), and distance addition map have been implemented to represent distance map information with GCS networks.

In distance map, the mean distance between the synaptic vector of each neuron and the synaptic vectors of all its direct neighbours is calculated. U-map represents the same information than distance map but, in addition it includes the distance between all the neighbouring neurons (painted in a circle form between each pair of neighbour units). Finally, the sum of the distance between the synaptic vector of a neuron and the synaptic vectors of the rest of units is calculated, when distance addition map is generated. In distance map and U-map, dark zones represent clusters and clear zones boundaries along with them. In distance addition map, neurons with near synaptic vectors appear with similar colour, and boundaries can be detected analyzing the regions where a considerable colour

variation exists. Using GCS networks, separated meshes represent different input clusters, usually. Fig. 3 shows an example of these three graphs, compared with the traditional Kohonen's maps, when an eleven separate regions distribution data set is used. GCS network represents eleven clusters in the three graphs, clearly. Distance map and U-map in Kohonen's network show the eleven clusters too, but in distance addition map it is not possible to distinguish them.

Displaying Projections

This technique takes into account the input distribution patterns to generate different values to assign to each neuron. For GCS networks, data histograms and quantization error maps have been implemented.

Generating the histogram, the number of training patterns associated to each neuron is obtained. However, when quantization error graph has to be produced, the sum of the distances between the synaptic vector of a neuron and the input vectors that lies in its *Voronoi* region is calculated. In both graphs, dark and clear zones correspond with high and low probability density areas, respectively, so it can be used in cluster analysis. Fig. 4 shows an example of these two methods compared with those obtained using Kohonen's model when chain-link distribution data set is used. Using Kohonen's model is difficult to distinguish the number of clusters present in the input space. On the other hand, GCS model has generated three output meshes, two of them representing one ring.

Figure 3. From left to right: distance map, U-map (unified distance map), and distance addition map when an eleven separate regions distribution data set is used. (a) Kohonen feature map with 10x10 grid of neurons. (b) GCS network with 100 output neurons. The right column shows the input data and the network projection using the two component values of the synaptic vectors.
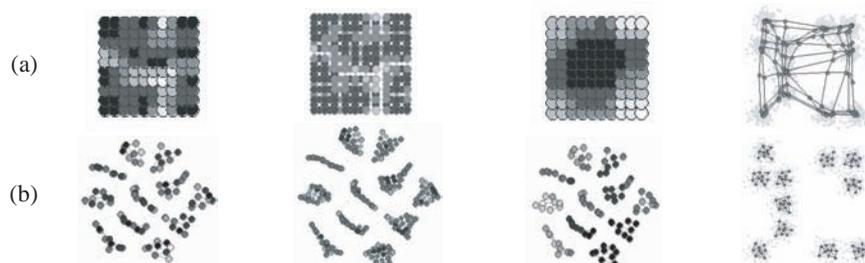
Displaying Components

The displaying components technique analyzes each synaptic vector or reference vector component in an individual manner. This kind of graphs offers a visual analysis of the topology preserving of the network, and a possible detection of correlations and dependences between training data components. Direct visualization of synaptic vectors and component planes graphs have been implemented for GCS networks.

Direct visualization map represents each neuron in a circle form within its synaptic vector inside in a graphical manner. This graph can be complemented with anyone of described in the previous sections, enriching its interpretation. A component plane map visualizes an individual component of all the synaptic vectors.

When all the component planes are generated, relations between weights can be appreciated if similar structures appear in identical places of two different component planes. Fig. 5 shows an example of these two displaying methods when multi-band data of satellite image is used. The direct visualization map shows the similarity between neighbouring units synaptic vectors, and, it is interesting distinguish the fact that all the neurons in a cluster have similar synaptic shapes. Furthermore, the integrated information about the distance addition map shows that there is no significant colour variation inside the same cluster. The six component

Figure 4. From left to right: Unified distance map, data histograms and quantization error maps when chain-link distribution data set is used. (a) Kohonen feature map with 10x10 grid of neurons. (b) GCS network with 100 output neurons. The right column shows the input data and the network projection using the three component values of the synaptic vectors.
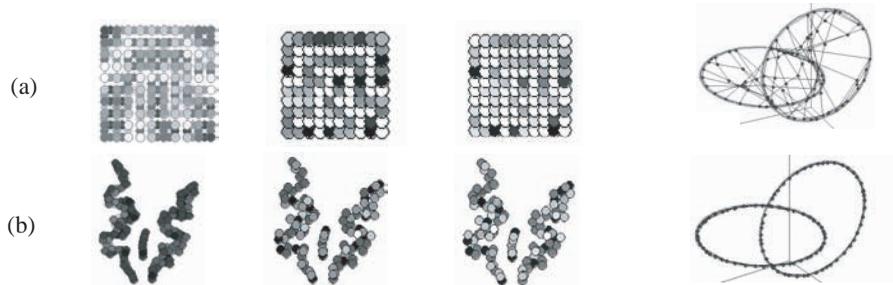
Figure 5. GCS network trained with multidimensional data of satellite image, 54 output neurons. Graphs from (a) to (f) show the component planes for the six elements of the synaptic vectors. (g) Direct visualization map using distance addition map additional information.
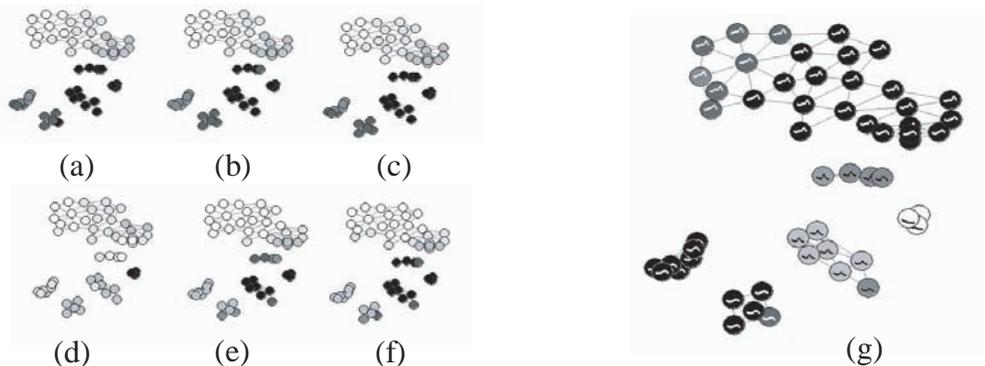
plane graphs exhibit possible dependences involving TM1, TM2 and TM3 input vector components and, TM5 and TM7 components too.

Results

Several Kohonen and GCS networks have been trained in order to evaluate and compare the resulting visualization graphs. For the sake of space only a few of these maps have been included here. Fig. 3 and Fig. 4 compare Kohonen and GCS visualizations using distance map, U-map, distance addition map, data histograms and quantization error map. It can be observed that GCS model offers much better graphical results in clusters analysis than Kohonen networks. The removal of units and connections inside low probability distribution areas causes that GCS network presents within a particular cluster the same quality of information that Kohonen network in relation to the entire map. Since it has already been mentioned, the grey scale used in all the maps is relative to the maximum and minimum values taken by the studied parameter. In all the cases the range of values taken by the calculated factor using GCS is minor than using Kohonen maps.

The exposed visualization methods applied to the visual analysis of multidimensional satellite data has given very satisfactory results (Fig 5). All trained GCS networks have been able to generate six sub maps in the output layer (in some case they have arrived up to eight) that identify the six land cover classes present in the sample of data. The direct visualization map and the component plane graphs have demonstrated to be a useful tool for the extraction of knowledge of the multisensorial data.

FUTURE TRENDS

The proposed knowledge visualization method based on GCS networks has results a useful tool for multidimensional data analysis. In order to evaluate the quality of the trained networks we consider necessary to develop some measure techniques (qualitative and quantitative in numerical and graphical format) to analyze the topology preservation obtained. In this way we will be able to validate the information visualized by the methods presented in this paper.

Also it would be interesting to validate these methods of visualisation with new data sets of very high

dimensional nature. We need to study the viability of cluster analysis with this projection technique when this class of data samples is used.

CONCLUSION

The exposed embedding method allows multidimensional data to be displayed as two-dimensional grey images. The visual-spatial abilities of human observers can explore these graphical maps to extract interrelations and characteristics in the dataset.

In GCS model the networks size does not have to be specified in advance. During the training process, the size of the network grows and decreases adapting its architecture to the particular characteristics of the training dataset.

Although in GCS networks it is necessary to determine a great number of training factors than in Kohonen model, using the learning modified model the tuning of the training factors values is simplified. In fact, several experiments have been made on datasets of diverse nature using the same values for all the training factors and giving excellent results in all the cases.

Especially notable is the cluster detection during the self-organization process without any other additional information.

REFERENCES

Delgado S., Gonzalo C., Martínez E., & Arquero A. (2004). Improvement of Self-Organizing Maps with Growing Capability for Goodness Evaluation of Multispectral Training Patterns. *IEEE International Proceedings of the Geoscience and Remote Sensing Symposium.* 1, 564-567.

Franzmeier M., Witkowski U., & Rückert U. (2005). Explorative data analysis based on self-organizing maps and automatic map analysis. *Lecture Notes in Computer Science.* 3512, 725-733.

Fritzke, B (1994). Growing Cell Structures – A self-organizing Network for Unsupervised and Supervised Learning. *Neural Networks.* 7(9), 1441-1460.

Fritzke, B (1995). A growing neural gas network learns topologies. *Advances in neural information processing systems.* 7, 625-632.

G

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*. (43), 59-69.

Kohonen, T. (2001). *Self-Organizing Map (*3rd ed). Springer, Berlin Heidelberg New York.

Kraaijveld MA., Mao J., & Jain AK. (1995). A non linear projection method based on Kohonen's topology preserving maps. *IEEE Transactions on Neural Networks. 6*(3), 548-559.

Merlk D., & Rauber A. (1997). Alternative ways for cluster visualization in self-organizing maps. *Workshop on Self-Organizing Maps*, Helsinki, Finland.

Rossi, F. (2006). Visual data mining and machine learning. *Proceedings of the European Symposium on Artificial Neural Networks*, Bruges, Belgium. 251-264.

Rubio M., & Giménez V. (2003). New methods for self-organizing map visual analysis. *Neural Computation & Applications*. 12, 142-152.

Tukey, JW. (1977). *Exploratory data analysis*. Addison-Wesley, Reading, MA.

Ultsch A., & Siemon HP. (1990). Kohonen self-organizing feature maps for exploratory data analysis. *Proceedings of the International Neural Network*, Dordrecht, The Nederlands.

Vesanto, J. (1999). SOM-based visualization methods. *Intelligent Data Analysis*, Elsevier Science, 3(2), 111-126.

Walker AJ., Cross SS., & Harrison RF. (1999). Visualisation of biomedical datasets by use of growing cell structure networks: a novel diagnostic classification technique. *The Lancer*, Academic Research Library. 1518-1521.

KEY TERMS

Artificial Neural Networks: An interconnected group of units or neurons that uses a mathematical model for information processing based on a connectionist approach to computation.

Data Mining: The application of analytical methods and tools to data for the purpose of identifying patterns, relationships or obtaining systems that perform useful tasks such as classification, prediction, estimation, or affinity grouping.

Exploratory Data Analysis: Philosophy about how a data analysis should be carried out. Exploratory data analysis employs a variety of techniques (mostly graphical) to extract the knowledge inherent to the data.

Growing Cell Structures: Growing variant of the self-organizing map model, with the peculiarity of dynamically adapts the size and connections of the output layer to the characteristics of the training patterns.

Knowledge Visualization: The creation and communication of knowledge through the use of computer and non-computer-based, complementary, graphic representation techniques.

Self-Organizing Map: A subtype of artificial neural network. It is trained using unsupervised learning to produce low dimensional representation of the training samples while preserving the topological properties of the input space

Unsupervised Learning: Method of machine learning where a model is fit to observations. It is distinguished from supervised learning by the fact that there is no a priori output.

GTM User Modeling for aIGA Weight Tuning in TTS Synthesis

Lluís Formiga
Universitat Ramon Llull, Spain

Francesc Alías
Universitat Ramon Llull, Spain

INTRODUCTION

Unit Selection Text-to-Speech Synthesis (**US-TTS**) systems produce synthetic speech based on the **retrieval** of previous recorded speech units from a **speech database (corpus)** driven by a weighted cost function (Black & Campbell, 1995). To obtain high quality synthetic speech these **weights must be optimized** efficiently. To that effect, in previous works, a technique was introduced for **weight tuning** based on evolutionary perceptual tests by means of **Active Interactive Genetic Algorithms (aiGAs)** (Alías, Llorà, Formiga, Sastry & Goldberg, 2006) **aiGAs** mine models that map **subjective preferences** from users by **partial ordering graphs**, **synthetic fitness** and **Evolutionary Computation (EC)** (Llorà, Sastry, Goldberg, Gupta & Lakshmi, 2005). Although **aiGA** propose an effective method to **map *single* user preferences**, as far as we know, the methodology to extract common solutions among different **individual preferences** (hereafter denoted as *common knowledge*) has not been tackled yet. Furthermore, there is an ambiguity problem to be solved when different users evolve to different weight configurations. In this review, **Generative Topographic Mapping (GTM)** is introduced as a method to extract common knowledge from **aiGA** models obtained from **user preferences.**

BACKGROUND

Weight Tuning in Unit-Selection Text-to-Speech Synthesis

The aim of **US-TTS** is to generate synthetic speech by concatenating the sequence of units that best fit the requirements derived from the input text. The speech units are retrieved from a **database** (**speech corpus**) which stores speech-units previously recorded by a professional speaker, typically.

Text-to-speech workflow is generally modelled as two independent blocks that convert written text into speech signal. The first block is named Natural Language Processing (NLP), which is followed by the Digital Signal Processing block (DSP). At first stage, The NLP block carries out a text preprocessing (e.g. conversion of digit numbers or acronyms to words), then it converts graphemes to phonemes. And at last stage, the NLP block assigns quantified prosody parameters to each phoneme guiding the way each phoneme is converted to signal. Generally, this quantified prosody parameters involve duration, pitch and energy. Next, The DSP block retrieves from a recorded **database (speech corpus)** the sequence of units that best matches the target requirements (the phonemes and their prosody). Finally, the speech units are ensembled to obtain the output speech signal.

The **retrieval** process is done by a dynamic programming algorithm (e.g. Viterbi or A* (Formiga & Alías, 2006)) driven by a cost function. The cost function computes the load of **selecting** a **unit** within a sequence as the sum of two weighted subcosts (see equation (1)): the target subcost (C^t) and the concatenation subcost (C^c). In this work, the C^t is considered as a weighted linear combination of the normalized prosody distances between the target-NLP predicted prosody vector and the candidate unit prosody vector (see equation). Otherwise, the C^c is computed as a weighted linear combination of the distances between the feature vectors of the speech signal around its concatenation point (see equation).

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=2}^{n} C^c(u_{i-1}, u_i)$$

(1)

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w_j^t C_j^u(t_i, u_i) \tag{2}$$

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \tag{3}$$

where t_1^n represents the target units sequence $\{t_1, t_2, ..., t_n\}$ and u_1^n represents the candidate units sequence $\{u_1, u_2, ..., u_n\}$.

$$C_j^t(t_i, u_i) = 1 - e^{\left(\frac{P_j(t_i)}{\sigma_{P_j}}\right)^2} \tag{4}$$

$$C_j^c(u_{i-1}, u_i) = 1 - e^{\left(\frac{P_j^R(u_{i-1}) - P_j^L(u_i)}{\sigma_{P_j}}\right)^2} \tag{5}$$

Appropriate design of cost function by means of **weight training** is a crucial to earn high quality synthetic speech (Black, 2002). Nevertheless this concern has focused approaches with no unique response. Several techniques have been suggested for **weight tuning**, which may be spitted into three families: *i)* manual-tuning *ii)* computationally-driven purely objective methods and *iii)* perceptually optimized techniques (Alías, Llorà, Formiga, Sastry & Goldberg, 2006). The present review is based on the techniques based on human feedback to the training process, following previous work (Alías, Llorà, Formiga, Sastry & Goldberg, 2006), which is outlined in the next section.

The Approach: Interactive Evolutionary Weight Tuning

Computationally-driven purely objective methods are mainly focused on an acoustic measure (obtained from cepstral distances) between the resynthesized and the natural signals. Hunt and Black adopted two approaches in (Hunt & Black, 1996). The first approach was based on **adjusting the weights** through an exhaustive search of a prediscretized weight space (weight space search, WSS). The second approach proposed by the authors used a multilinear regression technique (MLR), across the entire **database** to **compute the desired weights**. Later, Meron and Hirose (Meron & Hirose, 1999) presented a methodology that improved the efficiency of the WSS and refined the MLR method. In a previous

work (Alías & Llorà, 2003), introduced **evolutionary computation** to perform this tuning. More precisely, Genetic Algorithms (GA) were applied to obtain the most appropriate weight. The main added value of making use of GA to find optimal weight configuration is the independency to linear search models (as in MLR)and, in addition, it avoids the exhaustive search (as in WSS).

However, all this methods lack on its dependency on the acoustic measure to determine the actual quality of the synthesized speech, which in most part is relative to human hearing. To obtain better speech quality, it was suggested that user should take part in the process. In (Alías, Llorà, Iriondo, Sevillano, Formiga & Socoró, 2004) there were conducted preference tests by synthesizing the training text according to two different weights and comparing the obtained speech subjective quality. Subsequently, **Active Interactive Genetic Algorithms** were presented in (Llorà, Sastry, Goldberg, Gupta & Lakshmi, 2005) as one interactive **evolutionary computation** method where the user feedback evolves the solutions through *survival-of-the-fittest* mechanism. The solutions inherent **fitness** is based on the **partial order** provided by the evaluator; **Active iGAs** base its efficiency on evolving different solutions by means of **surrogate fitness**, which generalize the **user preferences**. This surrogate fitness and the **evolutionary process** are based on the following key elements: *i)* **partial ordering**, *ii)* induced complete order, and *iii)* surrogate function via ε Support Vector Machines (ε-SVM). Preference decisions made by the user are modelled as a **directional graph** which is used to generate partial ordering of solutions (e.g: $\hat{x}_1 > \hat{x}_2; \hat{x}_2 > \hat{x}_3 : \hat{x}_1 \rightarrow \hat{x}_2 \rightarrow \hat{x}_3$) (see figure 1). Table 1 shows the approach of global rank based on dominance measure: given a vertex v, the number of dominated vertexes $\delta(v)$ and dominating vertexes is computed. Using this measures, the estimated fitness may be computed as $\hat{f}(v) = \delta(v) - (v)$. The estimated ranking $\hat{r}(v)$ is obtained by sorting based on $\hat{f}(v)$ (Llorà, Sastry, Goldberg, Gupta & Lakshmi, 2005). The procedure of **aiGA** is detailed in algorithm 1.

However, once the global weights were obtained with **aiGA**, there was no single dominant weight solution (Alías, Llorà, Formiga, Sastry & Goldberg, 2006), i.e. each test performed by different users gave similar and different solutions. This fact implied that a second group of users had to validate the obtained weights.

G

Figure 1. (Left) Partial evaluations allow building a directed graph among all solutions. (Right) Obtained graph must be cleared to avoid cycles and draws.
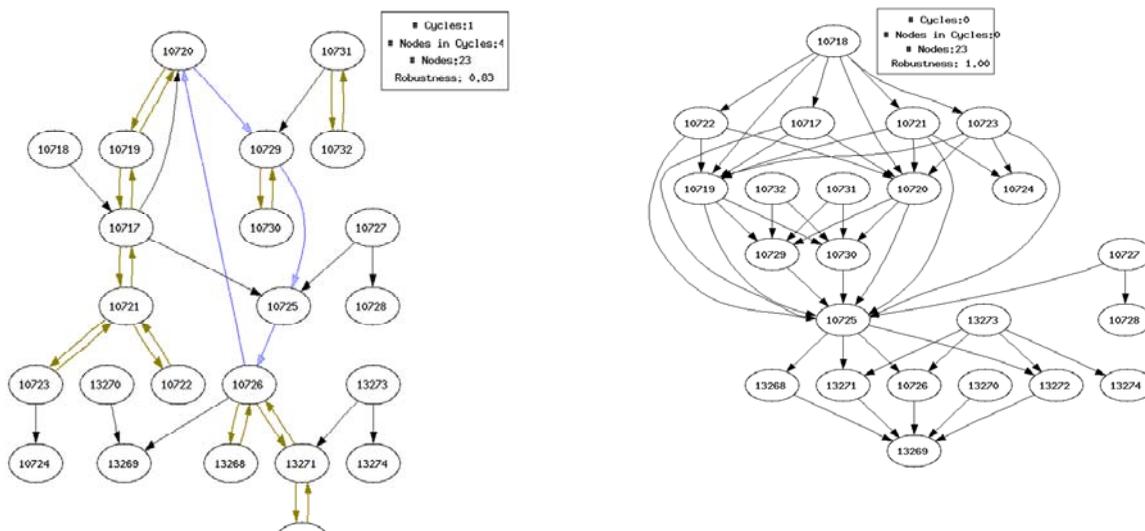
Table 1. Estimation of the global ranking based on the dominance measure. Dur T Ene T and Pit T stand for the weight values for target weights (duration, energy and pitch). In the same way, Pit C Ene C and Mfc C stand for the weight values for concatenation weights (Pitch, Energy and Mel Frequency Cepstrum).

v	$\delta(v)$	$\phi(v)$	$\hat{f}(v)$	Dur T	Ene C	Ene T	Mfc C	Pit C	Pit T	$\hat{r}(v)$	ε-SVM $\hat{f}(v)$
10718	15	0	15	0.04	0.32	0.04	0.27	0.22	0.12	1	(0.189)
10721	11	1	10	0.27	0.08	0.26	0.1	0.11	0.18	2.5	(3.294)
10723	11	1	10	0.17	0.16	0.05	0.01	0.31	0.29	2.5	(3.286)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
13271	1	14	-13	0.17	0.16	0.02	0.11	0.31	0.23	21	(13.091)
13272	1	14	-13	0.24	0.12	0.25	0.12	0.26	0.01	21	(20.174)
13269	0	19	-19	0.01	0.21	0.01	0.17	0.32	0.27	23	(23.792)

Thus, clustering problem from different tests was suitable to the **weight tuning** problem with the goal of extracting consistent results from the user tests.

GENERATIVE TOPOGRAPHIC MAPPING BEING A PLUS

GTM in a Nutshell

Unsupervised learning allows to group sparse data into clusters in terms of similarity of data samples. Several

Algorithm 1

Algorithm 1 Algorithm description of **active iGA**
procedure *aiGA*()
1 → Create an empty directed graph G .
2 → Create 2^h random initial solutions (S set).
3 → Create the hierarchical tournament set T using the available solutions in S .
4 → Present the tournaments in T to the user and update the **partial ordering** in G .
5 → Estimate $\hat{r}(v)$ for each $v \in S$.
6 → Train the surrogate ε -SVM **synthetic fitness** based on S and $\hat{r}(v)$.
7 → Optimize the surrogate ε -SVM **synthetic fitness** with cGA.
8 → Create the S' set with 2^{h-1} different solutions where $S \cap S' = \varnothing$, sampling out of the probabilistic model evolved by cGA.
9 → Create the hierarchical tournament set T' with $2^h - 1$ tournaments using 2^{h-1} solutions in S and 2^{h-1} solucions in S' .
10 → $S \Leftarrow S \cup S'$.
11 → $T \Leftarrow T \cup T'$.
12 → Go to 4 while not converged.

methods perform this grouping (Figuereido & Jain, 2002): Expectation Maximization (EM), k-means, Gaussian Mixture Models (GMM), Self Organizing Maps (SOM) and **Generative Topographic Mapping**, among others.

Techniques may be grouped, according to (Figuereido & Jain, 2002), into two types of formulation: *i)* model-based methods (e.g. GMM, EM, **GTM**) and *ii)* heuristic methods (e.g. k-means or hierarchical agglomerative methods). The number of sources generating the data is the differential propriety. Indeed, model-based methods suppose that the observations have been fashioned by one (arbitrarily chosen and unidentified) source of a set of alternative arbitrary sources. Therefore, inferring these tuned sources and mapping the source to each observation leads to a clustering of the set of observations. Otherwise, heuristic methods assume only one source for the observed data considering similar heterogeneity for them.

Self-Organizing Maps (or Kohonen maps) (Kohonen, 1990) are a clustering technique based on neural networks. The easiness of visualizing of multidimen-

sional data is the largely appropriate added value of SOM. In addition, **Generative Topographic Mapping** is a nonlinear latent variable model introduced in (Bishop, Svensen & Williams, 1998). **GTM** intends to give an substitute answer to SOM by means of overcoming its restrictions which are listed in (Kohonen, 2006): *i)* the absence of a cost function, *ii)* the lack of a theoretical basis for choosing learning rate parameter schedules and neighbourhood parameters to ensure topographic ordering, *iii)* the absence of any general proofs of convergence and *iv)* the fact that the model does not define a probability density.

GTM is based on a constrained mixture of GMM whose parameters can be tuned through EM algorithm. The handicap of heuristic based models is that there is not *a-priori* distribution of the centroids for each cluster. In **GTM**, the set of latent points is modelled as a grid. A circular gaussian distribution is a point in the grid with its equivalent correspondence, through a weighted non-linear basis functions, onto the multidimensional space. Thus, grid is shaped to wrap the data due to the explicit order among the gaussian distributions

Modelling User Preferences by GTM

GTM is able to extract solutions from the different **aiGA** evolved graphs due to the consistency of its theoretical basis. The key objective is to recognize important clusters inside the evolved data space and therefore, determine the fitness entropy of each cluster in terms of fitness variance to choose the global weight configuration set.

GTM can model the best **aIGA** weights from multi-dimensional weight space into a two-dimensional space. Taking into account the cluster with higher averaged fitness and lower standard deviation allows selecting the best weight configuration from different user **aiGA** models. For adjusting this method the geometry of the gaussian distributions and the size of the latent space have to be set up manually. EM weights **GTM** centroids and the basis functions. Then, it is extracted from each cluster the average fitness as well as its standard deviation. The computation of the averaged fitness and standard deviation is computed from the set which its weight combinations bayesian *a posteriori* probability is the highest to the cluster.

It is to note that the **fitness** itself does not get involved into the optimization EM part on behalf it is relative to each user and is not known for unevaluated weight combinations for one specific user (unless ε-SVM predicted).

Experiments and Results

On (Formiga & Alías, 2007) common knowledge was extracted from user **evolved weights** from previous tests conducted on catalan **speech corpus** with 9863 recorded units (1207 diphones and triphones) (obtained from 1520 sentences and words) (Alías, Llorà, Formiga, Sastry & Goldberg., 2006). On that test, five phonetically balanced sentences where extracted from the **corpus** to perform the global **weight tuning** process by a web interface named SinEvo (Alías, Llorà, Iriondo, Sevillano, Formiga & Socoró, 2004). The evolved weights were normalized through Max-Min normalization to range all weights between 0 and 1. That test was conducted by three users, obtaining fifteen different weight configurations.

Figure 2. Performance of GTM: Different pareto fronts were analyzed for each configuration

G
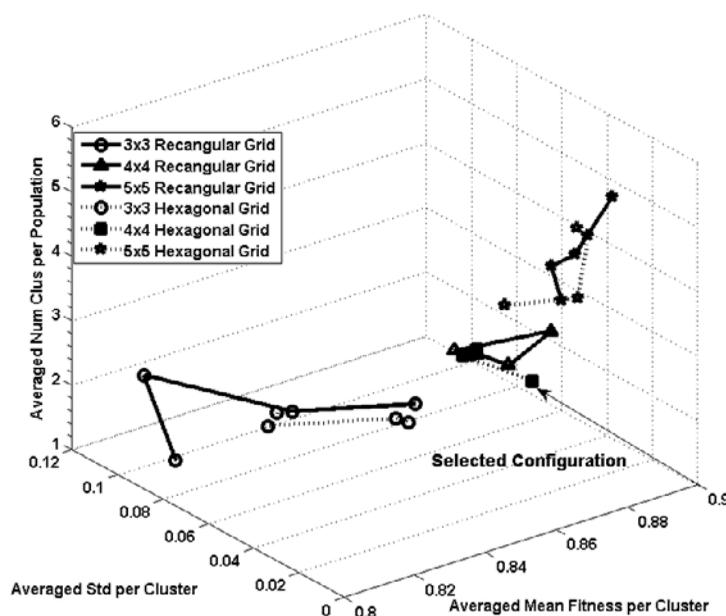
On (Formiga & Alías, 2007), different configurations of **GTM** were analyzed for **mapping normalized weights** (hexagonal or rectangular grid and different grid sizes: (3×3, 4×4, 5×5)). The purpose of this analysis was to find the optimal **GTM** configuration, i.e. The one which minimizes averaged standard deviation (std) per cluster and the number of significant clusters per population (with averaged fitness over 75%) while maximizing the averaged mean fitness per cluster. As it may be noticed in figure 2, the 4×4 grid configuration with hexagonal latent grid was selected as it yielded the best Pareto front (although the rest of 4x4 grids achieved similar performance).

After **GTM** was set up, each evolved weights were extracted and mapped to other users **GTM**s within the same sentence, obtaining their corresponding fitness from the other **users preferences**. Equation 6 allowed to set a global fitness (gF) from overall averaged fitness (F_{Av}^{GTM}) for each evolved weight configuration.

$$gF_{w_i} = \frac{1}{N} \cdot \sum_{i=1}^{U} \sum_{j=1}^{W} F_{Av}^{GTM}(w_j, u_i)$$

(6)

where U stands for the number of users, W for the number of weight configurations, N stands for the total number of weights ($U + W$).

In addition, to avoid a perceptual manual validation stage ten different users-not involved whatsoever in the tuning process-performed a comparison to the aIGA best weights to allow a comparison between **GTM** clustering and real human preference on validation stage.

Analyzing the results on figure 3, the **GTM** most voted weights configurations fit in with the manual **user preferences** for three sentences (*De la Seva Selva*, *Del Seu Territori* and *Grans Extensions*). Though, the rest of the sentences have quite different behaviour. The best weight combination selected from the users was the second **GTM** best weight configuration in *I els han venut* while the best **GTM** weight combination was never been voted. Cosine correlation is taken into account among problematic weights configurations as the important matter is weight distribution instead of analyzing the values themselves. In this case, **GTM** two better weights have a 0.7841 correlation, so **GTM** results may be measured satisfactory as weights approach equivalent patterns. By the other hand, the correlation

Figure 3. The results of the comparison between normalized user voted preferences and GTM mapping are presented for the five sentences. Two different solutions for same user were considered if they adopted similar fitness in aIGA model.
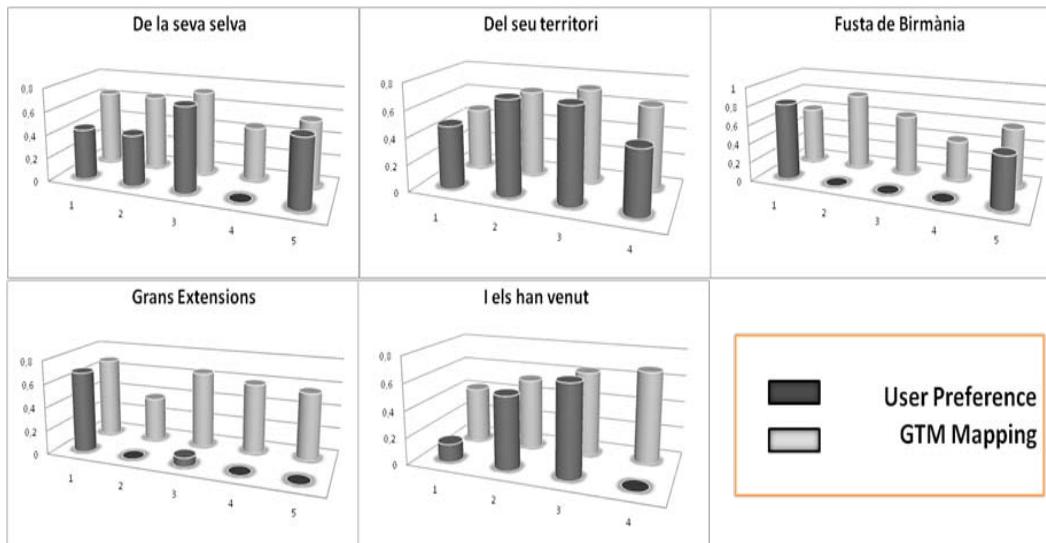
between the two best **GTM** weights configurations is 0.8323 in *Fusta de Birmània* and, as in the previous case, the correlation gives again satisfactory results.

FUTURE TRENDS

Future work will be focused on conducting new experiments, e.g. by clustering similar units instead of tackling global **weight tuning** on preselected sentences or by including more users in the training process. In addition the expansion of the capabilities of **GTM** to **map user preferences** opens the possibility to focus on non-linear cost functions so as to overcome the linearity restrictions of the present function.

CONCLUSIONS

This article continues the work of including **user preferences** for tuning the weights of the cost function in **Unit-selection TTS** systems. In previous works we have presented a method to find cost function optimal weight tuning based on perceptual criteria of individual users. As a next step, this paper applies a heuristic method for choosing the best solution among all users overcoming the need to conduct a second listening test to select the best weight configuration among individual optimal solutions. This proof-of-principle study shows that **GTM** is capable of **mapping the common knowledge** among different users thanks to working on the perceptually optimized weights space obtained through **aiGA** and getting a final solution that can be used for a final adjustment of the **TTS**.

REFERENCES

Alías, F., Llorà, X., Formiga, L., Sastry, K., Goldberg, D.E. (2006): Efficient interactive weight tuning for **tts** synthesis: reducing user fatigue by improving user consistency. In: *Proceedings of ICASSP. Volume I.*, Toulouse, France 865–868.

Alías, F., Llorà, X., Iriondo, I., Sevillano, X., Formiga, L., Socoro, J. C. (2004): Perception- Guided and Phonetic Clustering Weight Tuning Based on Diphone Pairs for **Unit Selection TTS**. In: *Proceedings of the*

8th International Conference on Spoken Language Processing (ICSLP) Jeju Island, Korea.

Alías, F., Llorà, X. (2003): Evolutionary weight tuning based on diphone pairs for **unit selection speech synthesis**. In: *Proceedings of the 8th European Conference on Speech Communication and Technology (EuroSpeech)*.

Bishop, C.M., Svensen, M., Williams, C.K.I. (1998): GTM: The generative topographic mapping. *Neural Comp.* 10(1) 215–234.

Black, A.W., Campbell, N. (1995): Optimising selection of units from **speech databases** for concatenative synthesis. In: *Proceedings of EuroSpeech. Volume 1.*, Madrid 581–584.

Black, A.W. (2002): for all of the people all of the time. In: *IEEE Workshop on Speech Synthesis (Keynote)*, Santa Monica, USA.

Figueiredo, M.A.F., Jain, A.K. (2002): Unsupervised learning of finite mixture models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(3) 381–396.

Formiga L., Alías F. (2007): Extracting User Preferences by GTM for aiGA Weight Tuning in Unit Selection Text-to-Speech Synthesis , *International Workshop on Artificial Neural Networks* (IWANN07), pp. 654-661, ISBN 978-3-540-73006-4, June 2007, Donostia (Spain).

Formiga L., Alías F. (2006): Heuristics for implementing the A* algorithm for **unit selection TTS** synthesis systems , *IV Jornadas en Tecnología del Habla* (4JTH06), pp. 219-224, ISBN 84-96214-82-6, november, Zaragoza (Spain).

Holland J. (1975): Adaptation in Natural and Artificial Systems. Ann arbor, Univ. of Michigan Press.

Hunt, A., Black, A.W. (1996): **Unit selection** in a concatenative speech synthesis system using a large speech database. In: Proceedings of ICASSP. Volume 1., Atlanta, USA 373–376.

Kohonen, T. (2006): Self-Organizing Maps. Springer.

Kohonen, T. (1990): The self-organizing map. Proceedings of the IEEE 78(9) 1464–1480.

Llorà., X., Sastry, K., Goldberg, D.E., Gupta, A., Lakshmi, L. (2005): Combating user fatigue in IGAs: Partial ordering, support vector machines, and synthetic fitness. *Proceedings of Genetic and Evolutionary Computation Conference* 2005 (GECCO-2005) 1363–1371 note: (Also IlliGAL Report No. 2005009).

Meron, Y., Hirose, K. (1999): Efficient weight training for selection based synthesis. In: *Proceedings of EuroSpeech*. Volume 5., Budapest, Hungary 2319–2322.

KEY TERMS

Correlation: A statistical measurement of the interdependence or association between two or qualitative variables. A typical calculation would be performed by multiplying a signal by either another signal (cross-correlation) or by a delayed version of itself (autocorrelation).

Digital Signal Processing (DSP): DSP, or Digital Signal Processing, as the term suggests, is the processing of signals by digital means. The processing of a digital signal is done by performing numerical calculations.

Diphone: A sound consisting of two phonemes: one that leads into the sound and one that finishes the sound. e.g.: "hello" silence-h h-eh eh-l l-oe oe-silence.

Evolutionary Algorithms: Collective term for all variants of (probabilistic) optimization and approximation algorithms that are inspired by Darwinian evolution. Optimal states are approximated by successive improvements based on the variation-selection-paradigm.

Generative Topographic Mapping (GTM): It is a technique for density modelling and data visualisation inspired in SOM (see SOM definition).

Mel Frequency Cepstral Coefficients (MFCC): The MFCC are the coefficients of the Mel cepstrum. The Mel-cepstrum is the cepstrum computed on the Mel-bands (scaled to human ear) instead of the Fourier spectrum.

Natural Language Processing (NLP): Computer understanding, analysis, manipulation, and/or generation of natural language.

Pitch: Intonation measure given a time in the signal.

Prosody: A collection of phonological features including pitch, duration, and stress, which define the rhythm of spoken language.

Text Normalization: The process of converting abbreviations and non-word written symbols into words that a speaker would say when reading that symbol out loud.

Unit Selection Synthesis: A synthesis technique where appropriate units are retrieved from **large databases** of natural speech so as to generate synthetic speech.

Unsupervised Learning: Learning techniques that group instances without a pre-specified dependent attribute. Clustering algorithms are usually unsupervised methods for grouping data sets.

Self-Organizing Maps: Self-organizing maps (SOMs) are a data visualization technique which reduce the dimensions of data through the use of self-organizing neural networks

Surrogate Fitness: Synthetic fitness measure that tries to evaluate one evolutionary solution in the same terms as one perceptual user would

Handling Fuzzy Similarity for Data Classification

Roy Gelbard
Bar-Ilan University, Israel

Avichai Meged
Bar-Ilan University, Israel

INTRODUCTION

Representing and consequently processing fuzzy data in standard and binary databases is problematic. The problem is further amplified in binary databases where continuous data is represented by means of discrete '1' and '0' bits. As regards classification, the problem becomes even more acute. In these cases, we may want to group objects based on some fuzzy attributes, but unfortunately, an appropriate fuzzy similarity measure is not always easy to find. The current paper proposes a novel model and measure for representing fuzzy data, which lends itself to both classification and data mining.

Classification algorithms and data mining attempt to set up hypotheses regarding the assigning of different objects to groups and classes on the basis of the similarity/distance between them (Estivill-Castro & Yang, 2004) (Lim, Loh & Shih, 2000) (Zhang & Srihari, 2004). Classification algorithms and data mining are widely used in numerous fields including: social sciences, where observations and questionnaires are used in learning mechanisms of social behavior; marketing, for segmentation and customer profiling; finance, for fraud detection; computer science, for image processing and expert systems applications; medicine, for diagnostics; and many other fields.

Classification algorithms and data mining methodologies are based on a procedure that calculates a similarity matrix based on similarity index between objects and on a grouping technique. Researches proved that a similarity measure based upon binary data representation yields better results than regular similarity indexes (Erlich, Gelbard & Spiegler, 2002) (Gelbard, Goldman & Spiegler, 2007). However, binary representation is currently limited to nominal discrete attributes suitable for attributes such as: gender, marital

status, etc., (Zhang & Srihari, 2003). This makes the binary approach for data representation unattractive for widespread data types.

The current research describes a novel approach to binary representation, referred to as Fuzzy Binary Representation. This new approach is suitable for all data types - nominal, ordinal and as continuous. We propose that there is meaning not only to the actual explicit attribute value, but also to its implicit similarity to other possible attribute values. These similarities can either be determined by a problem domain expert or automatically by analyzing fuzzy functions that represent the problem domain. The added new fuzzy similarity yields improved classification and data mining results. More generally, Fuzzy Binary Representation and related similarity measures exemplify that a refined and carefully designed handling of data, including eliciting of domain expertise regarding similarity, may add both value and knowledge to existing databases.

BACKGROUND

Binary Representation

Binary representation creates a storage scheme, wherein data appear in binary form rather than the common numeric and alphanumeric formats. The database is viewed as a two-dimensional matrix that relates entities according to their attribute values. Having the rows represent entities and the columns represent possible values, entries in the matrix are either '1' or '0', indicating that a given entity (e.g., record, object) has or lacks a given value, respectively (Spiegler & Maayan, 1985).

In this way, we can have a binary representation for discrete and continuous attributes.

Table 1. Standard binary representation table

Entity ID	Regular Representation		Binary Representation							
	Marital Status	Height	S	M	D	W	1.55	1.56	1.60	1.84
1	Married	1.60	0	1	0	0	0	0	1	0
2	Divorced	1.55	0	0	1	0	1	0	0	0
3	Single	1.84	1	0	0	0	0	0	0	1
4	Widowed	1.56	0	0	0	1	0	1	0	0
5	Single	1.60	1	0	0	0	0	0	1	0

Table 1 illustrates binary representation of a database consists of five entities with the following two attributes: Marital Status (nominal) and Height (continuous).

- Marital Status, with four values: **S** (single), **M** (married), **D** (divorced), **W** (widowed).
- Heights, with four values: **1.55**, **1.56**, **1.60** and **1.84**.

However, practically, binary representation is currently limited to nominal discrete attributes only. In the current study, we extend the binary model to include continuous data and fuzzy representation.

Similarity Measures

Similarity/distance measures are essential and at the heart of all classification algorithms. The most commonly-used method for calculating similarity is the Squared Euclidean measure. This measure calculates the distance between two samples as the square root of the sums of all squared distances between their properties (Jain & Dubes, 1988) (Jain, Murty & Flynn, 1999).

However, these likelihood-similarity measures are applicable only to ordinal attributes and cannot be used to classify nominal, discrete, or categorical attributes, since there is no meaning in placing such attribute values in a common Euclidean space. A similarity measure, which applicable to nominal attributes and used in our research is the Dice (Dice 1945).

Additional binary similarity measures were developed and presented (Illingworth, Glaser & Pyle, 1983) (Zhang & Srihari, 2003). Similarities measures between the different attribute values, as proposed in

Zadeh (1971) model, are essential in the classification process.

In the current study we use similarities between entities and between entity's attribute values to get better classification. Following former reserches, (Gelbard & Spiegler, 2000) (Erlich, Gelbard & Spiegler, 2002), the current study also uses Dice measure.

Fuzzy Logic

The theory of Fuzzy Logic was first introduced by Lotfi Zadeh (Zadeh, 1965). In classical logic, the only possible *truth-values* are *true* and *false*. In Fuzzy Logic; however, more *truth-values* are possible beyond the simple true and false. Fuzzy logic, then, derived from fuzzy set theory, is designed for situations where information is inexact and traditional digital on/off decisions are not possible.

Fuzzy sets are an extension of classical set theory and are used in fuzzy logic. In classical set theory, membership of elements in relation to a set is assessed according to a clear condition; an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in relation to a set; this is described with the aid of a membership function $\mu \rightarrow [0,1]$. An element mapped to the value 0 means that the member is not included in the given set, '1' describes a fully included member, and all values between 0 and 1 characterize the fuzzy members. For example, the continuous variable "Height" may have three membership functions; stand for "Short", "Medium" and "Tall" categories. An object may belong to few categories in different membership degree, e.g 180 cm. height may belong to the "Medium" and "Tall" categories, in different

membership degree expressed by the range [0,1]. The membership degrees are returned from the membership functions. We can say that a man whose height is 180 cm. is "slightly medium" and a man whose height is 200 cm. is of "perfect tall" height.

Different membership functions might represent different membership degrees. Having several possibilities for membership functions is part of the theoretical and practical drawbacks in Zada's model. There is no "right way" to determine the right membership functions (Mitaim & Kosko, 2001). Thus, a membership function may be considered arbitrary and subjective.

In the current work, we make use of membership functions to develop the enhanced similarity calculation for use in classification of fuzzy data.

FUZZY SIMILARITY REPRESENTATION

Standard Binary Representation exhibits data integrity in that it is precise, and preserves data accuracy without either loss of information or rounding of any value. The mutual exclusiveness assumption causes the "isolation" of each value. This is true for handling discrete data values. However, in dealing with a continuous attribute, e.g. Height, we want to assume that height 1.55 is closer to 1.56 than to 1.60. However, when converting such values into a mutually exclusive binary representation (Table 1), we lose these basic numerical relations. Similarity measures between any pair with different attribute values is always 0, no matter how similar the attribute values are. This drawback makes the standard binary representation unattractive for representing and handling continuous data types.

Similarity between attribute values is also needed for nominal and ordinal data. For example, the color "red" (nominal value) is more similar to the color "purple" than it is to the color "yellow". In ranking (question-naires) a "1" satisfactory rank (ordinal variable) might be closer to the "2" rank than to the "5" rank.

The absence of these similarity "intuitions" are of paramount importance in classification and indeed may cause some inaccuracies in classification results.

The following sections present a model that adds relative similarity values to the data representation. This serves to empower the binary representation to better handle both continuous and fuzzy data and improves classification results for all attribute types.

Model for Fuzzy Similarity Representation

In standard binary representation, each attribute (which may have several values, e.g., color: red, blue, green, etc.) is a vector of bits where only one bit is set to "1" and all others are set to "0". The "1" bit stands for the actual value of the attribute. In the Fuzzy Binary Representation, the zero bits are replaced by relative similarity values.

The Fuzzy Binary Representation is viewed as a two-dimensional matrix that relates entities according to their attribute values. Having the rows represent entities and the columns represent possible values, entries in the matrix are fuzzy numbers in the range [0,1], indicating the similarity degree of specific attribute value to the actual one, where '1' means full similarity to the actual value (this is the actual value), '0' means no similarity al all and all other values means partial similarity.

The following example illustrates the way for creating the Fuzzy Binary Representation: Let's assume we have a database of five entities and two attributes represented in a binary representation as illustrated in Table1. The fuzzy similarities between all attribute values are calculated (next section describes the calculation process) and represented in a two-dimensional "Fuzzy

Table 2. Fuzzy similarity matrixes of the marital status and height attributes

Marital Status	S	M	D	W
S	1	0	0.6	0.8
M	0	1	0	0
D	0.6	0	1	0.5
W	0.8	0	0.5	1

Height	1.55	1.56	1.60	1.84
1.55	1	0.9	07	0
1.56	0.9	1	0.8	0
1.60	0.7	0.8	1	0
1.84	0	0	0	1

Similarity Matrix", wherein rows and columns stand for the different attributes' values, and the matrix cells contain the fuzzy similarity between the value pairs. The Fuzzy Similarity Matrix is symmetrical. Table 2 illustrates fuzzy similarity matrixes for Marital Status and Height attributes.

The Marital Status similarity matrix shows that the similarity between Single and Widow is "high" (0.8), while there is no similarity between Single and Married (0). The Height similarity matrix shows that the similarity between 1.56 and 1.60 is 0.8 ("high" similarity), while the similarity between 1.55 and 1.84 is 0 (not similar at all). These similarity matrixes can be calculated automatically, as is explained in the next section.

Now, the zero values in the binary representation (Table 1) are replaced by the appropriate similarity value (Table 2). For example, in Table 1, we will replace the zero-bit stands for Height 1.55 of the first entity, with the fuzzy similarity between 1.55 and 1.60 (the actual attribute value), as indicated in the Height fuzzy similarity matrix (0.7). Table 3 illustrates the fuzzy representation accepted after such replacements.

It should be noted that the similarities indicated in the fuzzy similarity table relate to the similarity between the actual value of the attribute (e.g. 1.60 in entity 1) and the other attributes' values (e.g. 1.55, 1.56 and 1.84).

Next, the fuzzy similarities, presented in decimal form, are converted into a binary format – the Fuzzy Binary Representation. The conversion should allow similarity indexes like Dice.

To meet this requirement, each similarity value is represented by N binary bits, where N is determined by the required precision. For one- tenth precision, 10 binary bits are needed, for one-hundredth precision, 100 binary bits are needed. For ten bits precision fuzzy similarity "0" will be represented by ten '0's, the fuzzy similarity "0.1" will be represented by nine '0' followed by one '1', the fuzzy similarity "0.2" will be represented by eight '0's followed by two '1's and so on till the fuzzy similarity "1" which will be represented by ten '1's. Table 4 illustrates the conversion from fuzzy representation (Table 3) to fuzzy binary representation.

The Fuzzy Binary Representation illustrated in Table 4 is suitable for all data types (discrete and continuous) and, with the new knowledge (fuzzy similarities values) it contains, a better classification is expected.

The following section describes the process for similarity calculations necessary for this type of Fuzzy Binary Representation.

Table 3. Fuzzy similarity table

ID	Marital Status				Height			
	S	M	D	W	1.55	1.56	1.60	1.84
1	0	1	0	0	0.7	0.8	1	0
2	0.6	0	1	0.5	1	0.9	0.7	0
3	1	0	0.6	0.8	0	0	0	1
4	0.8	0	0.5	1	0.9	1	0.8	0
5	1	0	0.6	0.8	0.7	0.8	1	0

Table 4. Fuzzy binary representation table

ID	Marital Status				Height			
	S	M	D	W	1.55	1.56	1.60	1.84
1	0000000000	1111111111	0000000000	0000000000	0001111111	0011111111	1111111111	0000000000
2	0000111111	0000000000	1111111111	0000011111	1111111111	0111111111	0001111111	0000000000
3	1111111111	0000000000	0000111111	0011111111	0000000000	0000000000	0000000000	1111111111
4	0011111111	0000000000	0000011111	1111111111	0111111111	1111111111	0011111111	0000000000
5	1111111111	0000000000	0000111111	0011111111	0001111111	0011111111	1111111111	0000000000

Fuzzy Similarity Calculation

Similarity calculation between the different attribute values is not a precise science, i.e., there is no one way to calculate it, just as there is no one way to develop membership functions in the Fuzzy Logic world.

We suggest determining similarities according to the **attribute type**. A domain expert should evaluate similarity for nominal attributes like "Marital Status". For example, Single, Divorced and Widowed are considered "one person", while Married is considered as "two people". Therefore, Single may be more similar to Divorced and Widowed than it is to Married. On the other hand "Divorced" is one that once was married, so may be it is more similar to Married than to single. In short, similarity is a **relative**, rather than an absolute measure, as there is hardly any known automatic way to calculate similarities for such attributes and therefore a domain expert is needed.

Similarity for ordinal data like satisfactory rank can be calculated in the same way as for nominal or continuous attributes depending on the nature of attributes' values. Similarity for continuous data like Height can be calculated automatically. Unlike nominal attributes, in continuous data there is an intuitive meaning to the "distance" between different values. For example, as regards the Height attribute, the difference between 1.55 and 1.56 is smaller than the distance between 1.55 and 1.70; therefore, the similarity is expected to be higher accordingly. For continuous data, an automatic method can be constructed, as showed, to calculate the similarities.

Depending on the problem domain, a continuous attribute can be divided into one or more fuzzy sets (categories), e.g., the Height attribute can be divided into three sets: Short, Medium and Tall. A membership function for each set can be developed.

The calculated similarities depend on the specified membership functions; therefore, they are referred to here as fuzzy similarities. The following algorithm can be used for similarity calculations of continuous data:

For each pair of attribute values (v1 and v2)
  For each membership function F
    Similarities (v1, v2) = 1 - distance between F(v1) and F(v2)
    Similarity (v1, v2) = Maximum of the calculated Similarities

Now that we have discussed both a model for Fuzzy Binary Representation and a way to calculate similarities, we will show the new knowledge (fuzzy similarities) added to the standard binary representation improve the similarity measures between different entities, as discussed in the next section.

COMPARING STANDARD AND FUZZY SIMILARITIES

In this section, we compare standard and fuzzy similarities. The similarities were calculated according to the Dice index for the example represented in Table 4.

Table 5 combines similarities of the different entities related to (a) Martial Status (nominal), to (b) Height (continuous) and to (c) both the Marital Status and Height attributes.

Several points and findings arise from the representations shown above (Table 5). These are briefly highlighted below:

1. In our small example, a nominal attribute (Marital Status) represented in standard binary representation cannot be used for classification. In contrast, the Fuzzy Binary Representation, with a large diversity of similarities results, will enable better classification. Grouping entities with a similarity that is equal to or greater than 0.7 yields a class of entities 2, 3, 4 and 5, which represent Single, Divorced and Widowed that belong to the set "one person".

2. For a continuous attribute (Height) represented in the standard binary representation, classification is not possible. In contrast, the Fuzzy Binary Representation with diversity in similarities results will, once again, enable better classification. Entities 1 and 5 have absolute similarity (1), since for the Height attribute they are identical. Entities 2 and 4 (similarity = 0.94) are very similar, since they represent the almost identical heights of 1.55 and 1.56, respectively. Classification based on these two entities is possible due to diversity of similarities.

3. The same phenomena presented for a single attribute (Marital Status or Height) exist also for the both attributes (Marital Status + Height) when are taking together. Similarity greater than 0.8 is

Table 5. Entities similarity

Entity ID		Marital Status		Height		(a) Similarity for Marital Status		(b) Similarity for Height		(c) Similarity for Marital Status + Height	
A	B	A	B	A	B	Standard	Fuzzy	Standard	Fuzzy	Standard	Fuzzy
1	2	M	D	1.60	1.55	0	0	0	0.86	0	0.53
1	3	M	S	1.60	1.84	0	0	0	0	0	0
1	4	M	W	1.60	1.56	0	0	0	0.88	0	0.54
1	5	M	S	1.60	1.60	0	0	1	1	0.5	0.59
2	3	D	S	1.55	1.84	0	0.75	0	0	0	0.41
2	4	D	W	1.55	1.56	0	0.72	0	0.94	0	0.84
2	5	D	S	1.55	1.60	0	0.75	0	0.86	0	0.81
3	4	S	W	1.84	1.56	0	0.89	0	0	0	0.50
3	5	S	S	1.84	1.60	1	1	0	0	0.5	0.57
4	5	W	S	1.56	1.60	0	0.89	0	0.88	0	0.88

used to group entities 2, 4 and 5, which represent "one person" around 1.56 meters height.

Two important advantages of the novel Fuzzy Binary Representation detailed in the current work over the standard binary representation are suggested: (1) It is practically suitable to all attribute types. (2) It improves classification results.

FUTURE TRENDS

The current work improves classification by adding new similarity knowledge to the standard representation of data. Further research can be conducted to calculate the interrelationship between the different attributes, i.e., the cross-similarities among attributes such as marital status and height. Understanding such interrelationships might further serve to refine the classification and data mining results.

Another worthwhile research direction is helping the human domain expert to get the "right" similarities, and thus choose the "right" membership functions. A Decision Support System may provide a way in which to structure the similarity evaluation of the expert and make his/her decisions less arbitrary.

CONCLUSION

In the current paper, the problems of representing and classifying data in databases were addressed. The focus was on Binary Databases, which have been shown in recent years to have an advantage in classification and data mining. Novel aspects for representing fuzziness were shown and a measure of similarity for fuzzy data was developed and described. Such measures are required, as similarity calculations are at the heart of any classification algorithm. Classification examples were illustrated.

The evaluating of similarity measures shows that standard binary representation is useless when dealing with continuous attributes for classification. Fuzzy Binary Representation reforms this drawback and results in promising classification based on continuous data attributes. In addition, adding fuzzy similarity was also shown to be useful for regular (nominal, ordinal) data to ensure better classification. Summarily, fuzzy representation improves classification results for all attribute types.

REFERENCES

Dice, L.R. (1945). Measures of the amount of ecological association between species. *Ecology*, 26(3), 297-302.

Erlich, Z., Gelbard, R. & Spiegler, I. (2002). Data Mining by Means of Binary Representation: A Model for Similarity and Clustering. *Information Systems Frontiers*, 4(2), 187-197.

Estivill-Castro, V. & Yang J. (2004). Fast and Robust General Purpose Clustering Algorithms. *Data Mining and Knowledge Discovery*, 8(2), 127-150.

Gelbard, R. & Spiegler, I. (2000). Hempel's raven paradox: a positive approach to cluster analysis. *Computers and Operations Research*, 27(4), 305-320.

Gelbard, R., Goldman, O. & Spiegler, I. (2007). Investigating Diversity of Clustering Methods: An Empirical Comparison", *Data & Knowledge Engineering*, 63(1), 155-166.

Illingworth, V., Glaser, E.L. & Pyle, I.C. (1983). Hamming distance. In, *Dictionary of Computing*, Oxford University Press, 162-163.

Jain, A.K. & Dubes, R.C. (1988). *Algorithms for Clustering Data*. Prentice Hall.

Jain, A.K., Murty, M.N. & Flynn, P.J. (1999). Data Clustering: A Review. *ACM Communication Surveys*, 31(3), 264-323.

Lim, T.S., Loh, W.Y. & Shih, Y.S. (2000). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Machine Learning*, 40(3), 203-228.

Mitaim, S. & Kosko, B. (2001). The Shape of Fuzzy Sets in Adaptive Function Approximation. *IEEE Transactions on Fuzzy Systems*, 9(4), 637-656.

Spiegler, I. & Maayan, R. (1985). Storage and retrieval considerations of binary data bases. *Information Processing and Management*, 21(3), 233-254.

Zadeh, L.A., (1965). Fuzzy Sets. *Information and Control*, 8(1), 338-353.

Zadeh, L.A., (1971). Similarity Relations and Fuzzy Ordering. *Information Sciences*, 3, 177-200.

Zhang, B. & Srihari, S.N. (2003). Properties of Binary Vector Dissimilarity Measures. In, *Proc. JCIS Int'l Conf. Computer Vision, Pattern Recognition, and Image Processing,* 26-30.

Zhang, B., & Srihari, S.N. (2004). Fast k-Nearest Neighbor Classification Using Cluster-based Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4), 525-528.

KEY TERMS

Classification: The partitioning of a data set into subsets, so that the data in each subset (ideally) share some common traits - often proximity according to some defined similarity/distance measure.

Data Mining: The process of automatically searching large volumes of data for patterns, using tools such as classification, association rule mining, clustering, etc.

Database Binary Representation: A representation where a database is viewed as a two-dimensional matrix that relates entities (rows) to attribute values (columns). Entries in the matrix are either '1' or '0', indicating that a given entity has or lacks a given value.

Fuzzy Logic: An extension of Boolean logic dealing with the concept of partial truth. Fuzzy logic replaces Boolean truth values (0 or 1, black or white, yes or no) with degrees of truth.

Fuzzy Set: An extension of classical set theory. Fuzzy set theory used in Fuzzy Logic, permits the gradual assessment of the membership of elements in relation to a set.

Membership Function: The mathematical function that defines the degree of an element's membership in a fuzzy set. Membership functions return a value in the range of [0,1], indicating membership degree.

Similarity: A numerical estimate of the difference or distance between two entities. The similarity values are in the range of [0,1], indicating similarity degree.

Harmony Search for Multiple Dam Scheduling

Zong Woo Geem
Johns Hopkins University, USA

INTRODUCTION

The dam is the wall that holds the water in, and the operation of multiple dams is complicated decision-making process as an optimization problem (Oliveira & Loucks, 1997). Traditionally researchers have used mathematical optimization techniques with linear programming (LP) or dynamic programming (DP) formulation to find the schedule.

However, most of the mathematical models are valid only for simplified dam systems. Accordingly, during the past decade, some meta-heuristic techniques, such as genetic algorithm (GA) and simulated annealing (SA), have gathered great attention among dam researchers (Chen, 2003) (Esat & Hall, 1994) (Wardlaw & Sharif, 1999) (Kim, Heo & Jeong, 2006) (Teegavarapu & Simonovic, 2002).

Lately, another metaheuristic algorithm, harmony search (HS), has been developed (Geem, Kim & Loganathan, 2001) (Geem, 2006a) and applied to various artificial intelligent problems, such as music composition (Geem & Choi, 2007) and Sudoku puzzle (Geem, 2007).

The HS algorithm has been also applied to various engineering problems such as structural design (Lee & Geem, 2004), water network design (Geem, 2006b), soil stability analysis (Li, Chi & Chu, 2006), satellite heat pipe design (Geem & Hwangbo, 2006), offshore structure design (Ryu, Duggal, Heyl & Geem, 2007), grillage system design (Erdal & Saka, 2006), and hydrologic parameter estimation (Kim, Geem & Kim, 2001). The HS algorithm could be a competent alternative to existing metaheuristics such as GA because the former overcame the drawback (such as building block theory) of the latter (Geem, 2006a).

To test the ability of the HS algorithm in multiple dam operation problem, this article introduces a HS model, and applies it to a benchmark system, then compares the results with those of the GA model previously developed.

BACKGROUND

Before this study, various researchers have tackled the dam scheduling problem using phenomenon-inspired techniques.

Esat and Hall (1994) introduced a GA model to the dam operation. They compared GA with the discrete differential dynamic programming (DDDP) technique. GA could overcome the drawback of DDDP which requires exponentially increased computing burden. Oliveira and Loucks (1997) proposed practical dam operating policies using enhanced GA (real-code chromosome, elitism, and arithmetic crossover). Wardlaw and Sharif (1999) tried another enhanced GA schemes and concluded that the best GA model for dam operation can be composed of real-value coding, tournament selection, uniform crossover, and modified uniform mutation. Chen (2003) developed a real-coded GA model for the long-term dam operation, and Kim et al. (2006) applied an enhanced multi-objective GA, named NSGA-II, to the real-world multiple dam system. Teegavarapu and Simonovic (2002) used another metaheuristic algorithm, simulated annealing (SA), to solve the dam operation problem.

Although several metaheuristic algorithms have been already applied to the dam scheduling problem, the recently-developed HS algorithm was not applied to the problem before. Thus, this article deals with the HS algorithm's pioneering application to the problem.

HARMONY SEARCH MODEL AND APPLICATION

This article presents two major parts. The first part explains the structure of the HS model; and the second part applies the HS model to a bench-mark problem.

Dam Scheduling Model Using HS

The HS model has the following formulation for the multiple dam scheduling.

Maximize the benefits obtained by hydropower generation and irrigation

Subject to the following constraints:

1. **Range of Water Release**: the amount of water release in each dam should locate between minimum and maximum amounts.
2. **Range of Dam Storage**: the amount of dam storage in each dam should locate between minimum and maximum amounts.
3. **Water Continuity**: the amount of dam storage in next stage should be the summation of the amount in current stage, the amount of inflow, and the amount of water release.

The HS algorithm starts with filling random scheduling vectors in the harmony memory (HM). The structure of HM for the dam scheduling is as follows:

$$\begin{bmatrix} R_1^1 & R_2^1 & \cdots & R_N^1 & Z(\mathbf{R}^1) \\ R_1^2 & R_2^2 & \cdots & R_N^2 & Z(\mathbf{R}^2) \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ R_1^{HMS} & R_2^{HMS} & \cdots & R_N^{HMS} & Z(\mathbf{R}^{HMS}) \end{bmatrix} \tag{1}$$

Each row stands for each solution vector, and each column stands for each decision variable (water release amount in each stage and each dam). At the end of each row, the objective function value locates. HMS (harmony memory size) is the number of solution vectors is HM.

Based on the initial HM, a new scheduling can be generated with the following function:

$$R_i^{NEW} \leftarrow \begin{cases} R_i, R_i^{MIN} \leq R_i \leq R_i^{MAX} & \text{w.p.} \quad p_1 \\ R_i(k) \in \{R_i^1, R_i^2, ..., R_i^{HMS}\} & \text{w.p.} \quad p_2 \\ R_i(k) + \Delta & \text{w.p.} \quad p_3 \\ R_i(k) - \Delta & \text{w.p.} \quad p_4 \end{cases} \tag{2}$$

where R_i^{NEW} is a new water release amount for decision variable i; the first row in the right hand side means that the new amount is chosen randomly from the total range; the second row means that the new amount is chosen from the HM; the third and fourth rows means that the new amount is certain unit (Δ) higher or lower

than the original amount $R_i(k)$ obtained from the HM. The summation of probability is equal to one ($p_1 + p_2 + p_3 + p_4 = 1$).

If the newly-generated vector, $\mathbf{R}^{NEW}$, is better than the worst harmony in the HM in terms of objective function, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

If the HS model reaches MaxImp (maximum number of function evaluations), computation is terminated. Otherwise, another new harmony (= vector) is generated by considering one of three above-mentioned mechanisms.

Applying HS to a Benchmark Problem

The HS model was applied to a popular multiple dam system as shown in Figure 1 (Wardlaw & Sharif, 1999).

The problem has 12 two-hour operating periods, and only dam 4 has irrigation benefit because outflows of other dams are not directed to farms. The range of water releases is as follows:

$$0.0 \leq R_1 \leq 3, \, 0.0 \leq R_2, R_3 \leq 4, \, 0.0 \leq R_4 \leq 7 \tag{3}$$

The range of dam storages is as follows:

Figure 1. Schematic of four dam system
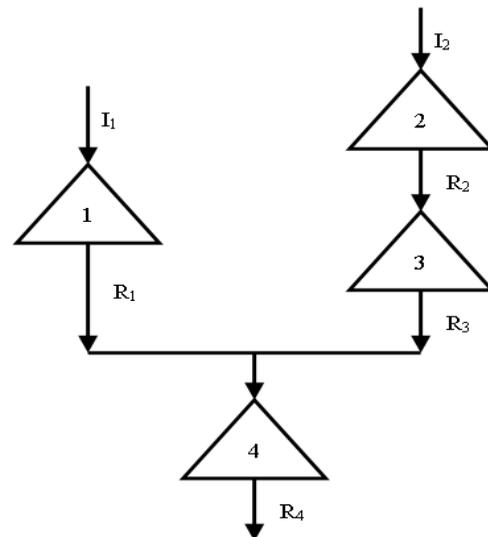
$$0.0 \leq S_1, S_2, S_3 \leq 10, 0.0 \leq S_4 \leq 15 \qquad (4)$$

The initial and final storage conditions are as follows:

$$S_1(0), S_2(0), S_3(0), S_4(0) = 5 \qquad (5)$$

$$S_1(12), S_2(12), S_3(12) = 5, S_4(12) = 7 \qquad (6)$$

H

Table 1. One example of optimal schedules by HS

Time	Dam 1	Dam 2	Dam 3	Dam 4
0	1.0	4.0	0.0	0.0
1	0.0	1.0	0.0	2.0
2	0.0	2.0	4.0	7.0
3	2.0	0.0	4.0	7.0
4	3.0	3.0	4.0	7.0
5	3.0	3.0	4.0	7.0
6	3.0	4.0	4.0	7.0
7	3.0	4.0	4.0	7.0
8	3.0	4.0	4.0	7.0
9	3.0	4.0	4.0	7.0
10	3.0	4.0	4.0	0.0
11	0.0	3.0	0.0	0.0

Figure 2. Water release trajectory in each dam

805
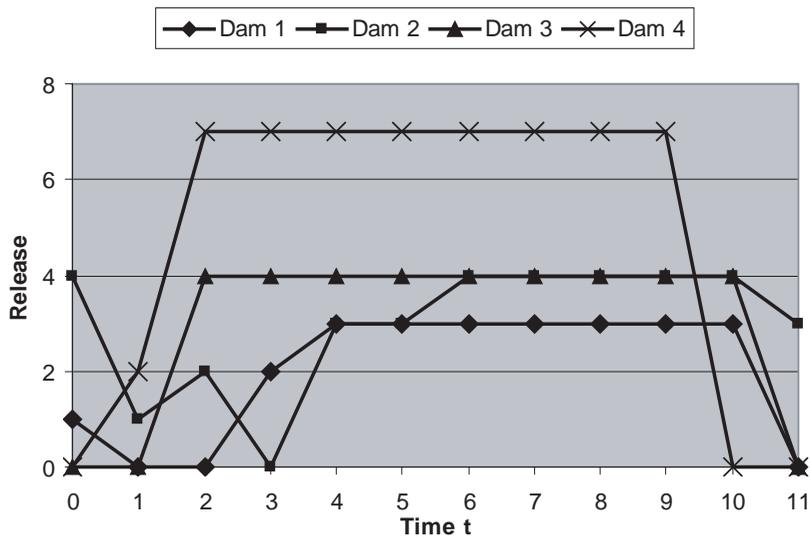
There are only two inflows: 2 units to dam 1; 3 units to dam 2.

$$I_1 = 2, I_2 = 3 \qquad (7)$$

Wardlaw and Sharif (1999) tackled this dam scheduling problem using an enhanced GA model (Population Size = 100; Crossover Rate = 0.70; Mutation Rate = 0.02; Number of Generations = 500; Number of Function Evaluations = 35,000; Binary, Gray, & Real-Value Representations; Tournament Selection; One-Point, Two-Point, & Uniform Crossovers; and Uniform and Modified Uniform Mutations). The GA model found a best near-optimal solution of 400.5, which is 99.8% of global optimum (401.3).

The HS model was applied to the same problem with the following algorithm parameters: HMS = 30; HMCR = 0.95; PAR = 0.05; and MaxImp = 35,000. The HS model could find five different global optimal solutions (HS1 ~ HS5) with identical cost of 401.3. Table 1 shows one example out of five optimal water release schedules.

Figure 2 shows corresponding release trajectories in all dams.

When the HS model was further tested with different algorithm parameter values, it found a better solution than that (400.5) of the GA model seven cases out of eight ones.

FUTURE TRENDS

From the success in this study, the future HS model should consider more complex dam scheduling problems with various real-world situations.

Also, algorithm parameter guidelines obtained from considerable experiments on the values will be helpful to engineers in practice because meta-heuristic algorithms, including HS and GA, require lots of trials to obtain best algorithm parameters.

CONCLUSION

Music-inspired algorithm, HS, was successfully applied to the optimal scheduling problem of the multiple dam system, outperforming the results of GA. While the GA model obtained near-optimal solutions, the HS model found five different global optima under the same number of function evaluations.

Moreover, the HS model did not perform sensitivity analysis of algorithm parameters while the GA model tested many parameter values and different operation schemes. This could reduce time and trouble in choosing parameter values in HS.

REFERENCES

Chen, L. (2003). Real Coded Genetic Algorithm Optimization of Long Term Reservoir Operation. *Journal of the American Water Resources Association*, 39(5), 1157-1165.

Erdal, F. & Saka, M. P. (2006). Optimum Design of Grillage Systems Using Harmony Search Algorithm. *Proceedings of the 5th International Conference on Engineering Computational Technology* (ECT 2006), CD-ROM.

Esat, V. & Hall, M. J. (1994). Water Resources System Optimization Using Genetic Algorithms. *Proceedings of the First International Conference on Hydroinformatics*, 225-231.

Geem, Z. W. (2006a). Improved Harmony Search from Ensemble of Music Players. *Lecture Notes in Artificial Intelligence*, 4251, 86-93.

Geem, Z. W. (2006b). Optimal Cost Design of Water Distribution Networks using Harmony Search. *Engineering Optimization*, 38(3), 259-280.

Geem, Z. W. (2007). Harmony Search Algorithm for Solving Sudoku. *Lecture Notes in Artificial Intelligence*, In Press.

Geem, Z. W. & Choi, J. Y. (2007). Music Composition Using Harmony Search Algorithm. *Lecture Notes in Computer Science*, 4448, 593-600.

Geem, Z. W. & Hwangbo, H. (2006). Application of Harmony Search to Multi-Objective Optimization for Satellite Heat Pipe Design. *Proceedings of US-Korea Conference on Science, Technology, & Entrepreneurship* (UKC 2006), CD-ROM.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, 76(2), 60-68.

Kim, J. H., Geem, Z. W., & Kim, E. S. (2001). Parameter Estimation of the Nonlinear Muskingum Model Using Harmony Search. *Journal of the American Water Resources Association*, 37(5), 1131-1138.

Kim, T., Heo, J. -H., Jeong, C. -S. (2006). Multireservoir System Optimization in the Han River basic using Multi-Objective Genetic Algorithm. *Hydrological Processes*, 20, 2057-2075.

Lee, K. S. & Geem, Z. W. (2004). A New Structural Optimization Method Based on the Harmony Search Algorithm. *Computers & Structures*, 82(9-10), 781-798.

Li, L., Chi, S. -C., & Chu, X. -S. (2006). Location of Non-Circular Slip Surface Using the Modified Harmony Search Method Based on Correcting Strategy. *Rock and Soil Mechanics*, 27(10), 1714-1718.

Oliveira, R., & Loucks, D. P. (1997). Operating Rules for Multireservoir Systems. *Water Resources Research*, 33(4), 839-852.

Ryu, S., Duggal, A.S., Heyl, C. N., & Geem, Z. W. (2007). Mooring Cost Optimization Via Harmony Search. *Proceedings of the 26th International Conference on Offshore Mechanics and Arctic Engineering*, ASME, CD-ROM.

Teegavarapu, R. S. V., Simonovic, S. P. (2002). Optimal Operation of Reservoir Systems Using Simulated Annealing. *Water Resources Management*, 16, 401-428.

Wardlaw, R., Sharif, M. (1999). Evaluation of Genetic Algorithms for Optimal Reservoir System Operation. *Journal of Water Resources Planning and Management*, ASCE, 125(1), 25-33.

KEY TERMS

Evolutionary Computation: Solution approach guided by biological evolution, which begins with potential solution models, then iteratively applies algorithms to find the fittest models from the set to serve as inputs to the next iteration, ultimately leading to a model that best represents the data.

Genetic Algorithm: Technique to search exact or approximate solutions of optimization or search problem by using evolution-inspired phenomena such as selection, crossover, and mutation. Genetic algorithm is classified as global search algorithm.

Harmony Search: Technique to search exact or approximate solutions of optimization or search problem by using music-inspired phenomenon (improvisation). Harmony search has three major operations such as random selection, memory consideration, and pitch adjustment. Harmony search is classified as global search algorithm.

Metaheuristics: Technique to find solutions by combining black-box procedures (heuristics). Here, 'meta' means 'beyond', and 'heuristic' means 'to find'.

Multiple Dam Scheduling: Process of developing individual dam schedule in multiple dam system. The schedule contains water release amount at each time period while satisfying release limit, storage limit, and continuity conditions.

Optimization: Process of seeking to optimize (minimize or maximize) an objective function while satisfying all problem constraints by choosing the values of continuous or discrete variables.

Soft Computing: Collection of computational techniques in computer science, especially in artificial intelligence, such as fuzzy logic, neural networks, chaos theory, and evolutionary algorithms.

H

Hierarchical Neuro–Fuzzy Systems Part I

Marley Vellasco
PUC-Rio, Brazil

Marco Pacheco
PUC-Rio, Brazil

Karla Figueiredo
UERJ, Brazil

Flavio Souza
UERJ, Brazil

INTRODUCTION

Neuro-fuzzy [Jang,1997][Abraham,2005] are **hybrid systems** that combine the learning capacity of **neural nets** [Haykin,1999] with the linguistic interpretation of **fuzzy inference systems** [Ross,2004]. These systems have been evaluated quite intensively in machine learning tasks. This is mainly due to a number of factors: the applicability of learning algorithms developed for **neural nets**; the possibility of promoting implicit and explicit knowledge integration; and the possibility of extracting knowledge in the form of fuzzy rules. Most of the well known **neuro-fuzzy systems**, however, present limitations regarding the number of inputs allowed or the limited (or nonexistent) form to create their own structure and rules [Nauck,1997][Nauck,1998][Vuorimaa,1994][Zhang,1995].

This paper describes a new class of **neuro-fuzzy models**, called Hierarchical Neuro-Fuzzy BSP Systems (HNFB). These models employ the BSP partitioning (**Binary Space Partitioning**) of the input space [Chrysanthou,1996] and have been developed to bypass traditional drawbacks of **neuro-fuzzy systems**. This paper introduces the HNFB models based on **supervised learning algorithm**. These models were evaluated in many benchmark applications related to **classification** and **time-series forecasting**. A second paper, entitled *Hierarchical Neuro-Fuzzy Systems Part II*, focuses on **hierarchical neuro-fuzzy models** based on reinforcement learning algorithms.

BACKGROUND

Hybrid Intelligent Systems conceived by using techniques such as Fuzzy Logic and Neural Networks have been applied in areas where traditional approaches were unable to provide satisfactory solutions. Many researchers have attempted to integrate these two techniques by generating hybrid models that associate their advantages and minimize their limitations and deficiencies. With this objective, hybrid neuro-fuzzy systems [Jang,1997][Abraham,2005] have been created.

Traditional neuro-fuzzy models, such as ANFIS [Jang,1997], NEFCLASS [Nauck,1997] and FSOM [Vuorimaa,1994], have a limited capacity for creating their own structure and rules [Souza,2002a]. Additionally, most of these models employ grid partition of the input space, which, due to the rule explosion problem, are more adequate for applications with a smaller number of inputs. When a greater number of input variables are necessary, the system's performance deteriorates.

Thus, *Hierarchical Neuro-Fuzzy Systems* have been devised to overcome these basic limitations. Different models of this class of **neuro-fuzzy systems** have been developed, based on **supervised technique**.

HIERARCHICAL NEURO-FUZZY SYSTEMS

This section presents the new class of **neuro-fuzzy systems** that are based on hierarchical partitioning.

Two sub-sets of **hierarchical neuro-fuzzy systems** (HNF) have been developed, according to the learning process used: **supervised learning models** (HNFB [Souza,2002b][Vellasco,2004], HNFB^{-1} [Gonçalves,2006], HNFB-Mamdani [Bezerra,2005]); and reinforcement learning models (RL-HNFB [Figueiredo,2005a], RL-HNFP [Figueiredo,2005b]). The focus of this paper is on the first sub-set of models, which are described in the following sections.

HIERACHICAL NEURO-FUZZY BSP MODEL

Basic Neuro-Fuzzy BSP Cell

An HNFB cell is a neuro-fuzzy mini-system that performs fuzzy **binary partitioning** of the input space. The HNFB cell generates a crisp output after a defuzzification process.

Figure 1(a) illustrates the cell's functionality, where '*x*' represents the input variable; $\rho(x)$ and $\mu(x)$ are the membership functions *low* and *high*, respectively, which generate the antecedents of the two fuzzy rules; and *y* is the crisp output. The linguistic interpretation of the mapping implemented by the HNFB cell is given by the following rules:

- If $x \in \rho$ then $y = d_1$
- If $x \in \mu$ then $y = d_2$.

Each rule corresponds to one of the two partitions generated by BSP. Each partition can in turn be subdivided into two parts by means of another HNFB cell.

The profiles of membership functions $\rho(x)$ and $\mu(x)$ are complementary logistic functions.

The output *y* of an HNFB cell (defuzzification process) is given by the weighted average. Due to the fact that the membership function $\rho(x)$ is the complement to 1 of the membership function $\mu(x)$, the following equation applies:

$$y = \rho(x)*d_1 + \mu(x)*d_2 \text{ or } y = \sum_{i=1}^{2}\alpha_i d_i \qquad (1)$$

where α_i symbolizes the firing level of the rule in partition *i* and are given by: $\alpha_1 = \rho(x)$; $\alpha_2 = \mu(x)$. Each d_i corresponds to one of the three possible consequents below:

- A singleton: The case where d_i = constant.
- A linear combination of the inputs:

$$d_i = \sum_{k=1}^{n} w_k x_k + w_0$$

where: x_k is the system's *k*-th input; the w_k represent the weight associated with the in put x_k; '*n*' is equal to the total number of inputs; and w_0 corresponds to a constant value.

- The output of a stage of a previous level: The case where $d_i = y_j$, where y_j represents the output of a generic cell '*j*', whose value is also calculated by eq. (1).

HNFB Architecture

An HNFB model may be described as a system that is made up of interconnections of HNFB cells. Figure 1(b) illustrates an HNFB system along with the respective partitioning of the input space. In this system, the initial partitions 1 and 2 ('BSP0' cell) have been subdivided; hence, the consequents of its rules are the outputs of BSP1 and BSP2, respectively. In turn, these subsystems have, as consequents, values d_{11}, y_{12}, d_{21} and d_{22}, respectively. Consequent y_{12} is the output of the 'BSP12' cell. The output of the system in figure 1(b) is given by equation (2).

$$y = \alpha_1(\alpha_{11}d_{11} + \alpha_{12}(\alpha_{121}d_{121} + \alpha_{122}d_{122})) + \alpha_2(\alpha_{21}d_{21} + \alpha_{22}d_{22})$$

$$(2)$$

It must be stressed that, although each BSP cell divides the input space only in two fuzzy set (*low* and *high*), the complete HNFB architecture divides the universe of discourse of each variable in as many partitions as necessary. The number of partitions is determined during the learning process. In Figure 1(c), for instance, the upper left part of the input space (partition 12 in gray) has been further subdivided by the horizontal variable x_1, resulting in three fuzzy sets for the complete universe of discourse of this specific variable.

Learning Algorithm

The HNFB system has a training algorithm based on the gradient descent method for learning the structure

Figure 1. (a) Interior of Neuro-Fuzzy BSP cell. (b) Example of HNFB system. (c) Input space Partitioning of the HNFB system

(a)   *(b)*   *(c)*
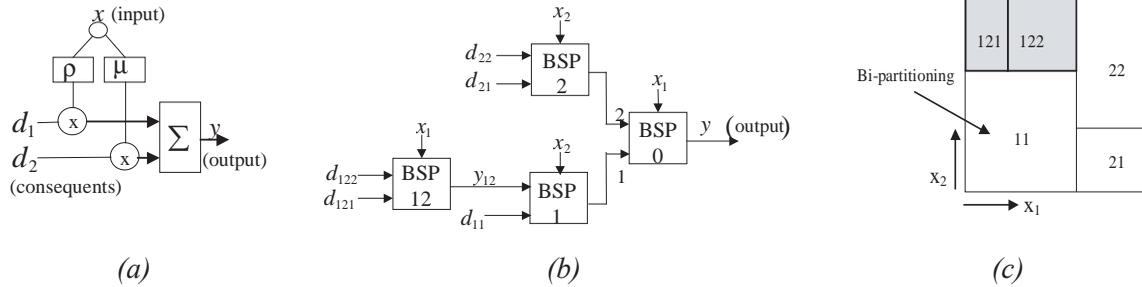
of the model and, consequently, linguistic rules. The parameters that define the profiles of the membership functions of the antecedents and consequents are regarded as fuzzy weights of the **neuro-fuzzy system**.

In order to prevent the structure from growing indefinitely, a non-dimensional parameter, named decomposition rate (δ), was created. More details of this algorithm may be found in [Souza,2002b][Gonçalves,2006].

The results obtained in **classification** and **time series forecasting** problems are presented in the Case Studies section.

HIERARCHICAL NEURO-FUZZY BPS FOR CLASSIFICATION

The original HNFB provides very good results for function approximation and **time series forecasting**. However, it is not ideal for **pattern classification** applications, since it has only one output and makes use of the **Takagi-Sugeno inference method** [Takagi,1985], which reduces the rule base interpretability.

Therefore, a new **hierarchical neuro-fuzzy** BSP **model** dedicated to **pattern classification** and **rule extraction**, called the Inverted HNFB or HNFB^{-1}, has been developed, which is able to extract classification rules such as: *If x is A and y is B then input-pattern belongs to class Z.* This new **hierarchical neuro-fuzzy model** is denominated *inverted* because it applies the learning process of the original HNFB to generate the model's structure. After this first learning phase, the

structure is inverted and the architecture of the HNFB^{-1} model is obtained. The basic cell of this new inverted structure is described below.

Basic Inverted-HNFB Cell

Similarly to the original HNFB model, a basic Inverted-HNFB cell is a neuro-fuzzy mini-system that performs fuzzy **binary partitioning** in a particular space according to the same membership functions ρ and μ. However, after a defuzzification process, the Inverted-HNFB cell generates two crisp outputs instead of one. Fig. 2(a) illustrates the interior of the Inverted-HNFB cell.

By considering that membership functions are complementary, the outputs of an HNFB^{-1} cell are given: $y_1 = \beta * \rho(x)$ and $y_2 = \beta * \mu(x)$, where β corresponds to one of the two possible cases below:

- β=**the input of the first cell**: so β =1.
- β=**is the output of a cell of a previous level**: so β=y_j, where y_j represents one of the two outputs of a generic '*j*' cell.

Inverted-HNFB Architecture

Fig. 2(b) presents an example of the original HNFB architecture obtained during the training phase of a database containing three distinct classes, while Fig. 2(c) shows how the HNFB^{-1} model is obtained, after the inversion process.

In the HNFB^{-1} architecture shown in Fig. 2(c), it may be observed that the classification system has several outputs (y_1 to y_5), one for each existing leaf in the original HNFB architecture. The outputs of the leaf cells are calculated by means of the following equations (using complementary membership functions):

$$y_1 = \rho_0 \cdot \rho_1 \tag{3}$$

$$y_2 = \rho_0 \cdot \mu_1 \cdot \rho_{12} \tag{4}$$

$$y_3 = \rho_0 \cdot \mu_1 \cdot \mu_{12} \tag{5}$$

$$y_3 = \rho_0 \cdot \mu_1 \cdot \mu_{12} \tag{6}$$

$$y_5 = \mu_0 \cdot \mu_2 \tag{7}$$

H

where ρ_i and μ_i are the membership functions for the BSP$_i$.

HNFB^{-1} System Outputs

After the inversion has been performed, the outputs are connected to T-conorm cells (OR operator) that define the classes (see Fig. 2(d)). The initial procedure for linking the leaf cells to the T-conorm neurons consists of connecting all leaf cells with all T-conorm neurons. Once these connections have been made, it is necessary to establish their weights. For the purpose of assigning

Figure 2. (a) HNFB^{-1} basic cell. (b) Original HNFB architecture. (c) Inversion of the architecture shown in Fig. 2(b). (d) Connection of the inverted architecture to T-conorm cells
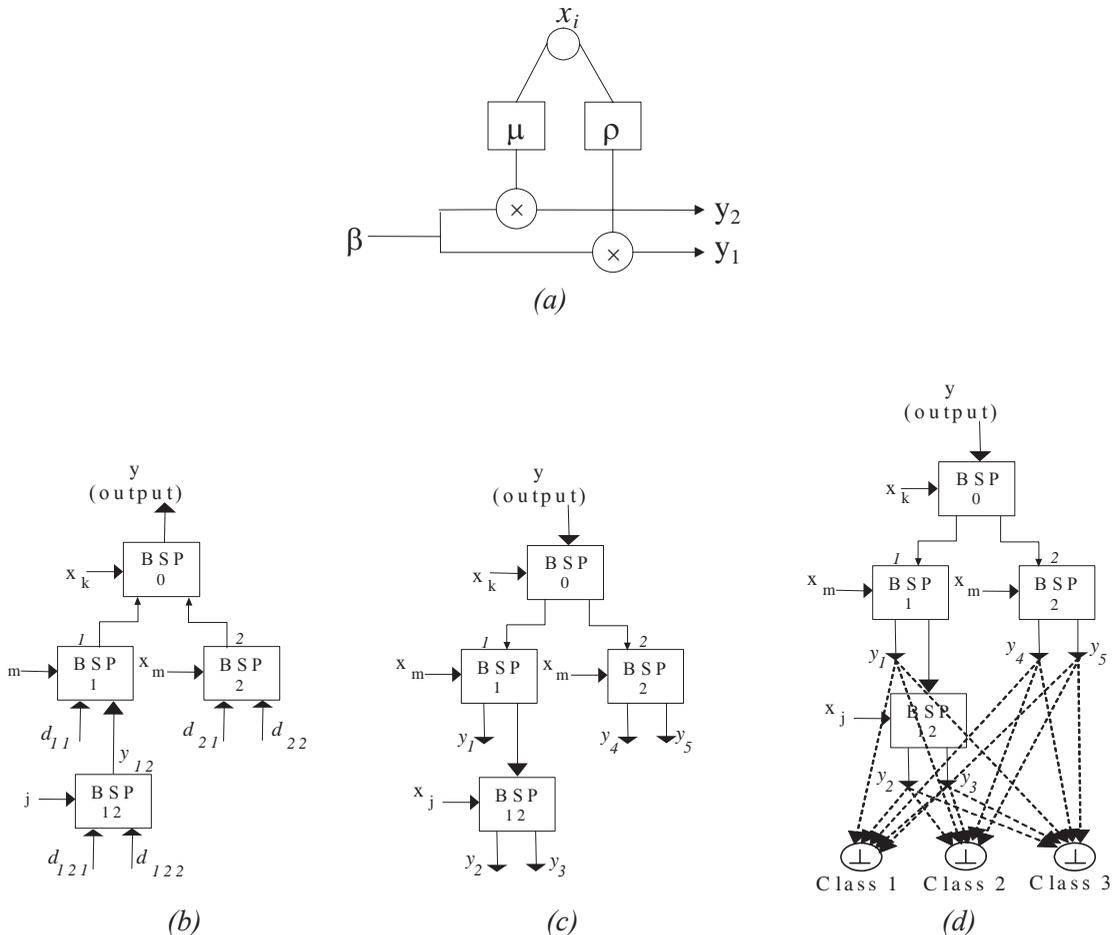
these weights, a learning method based on the Least Mean Squares [Haykin 1999] has been employed.

After the weights have been determined, the T-conorm operation (Limited Sum T-conorm operator [Ross,2004]) is used for processing the output of the neuron. The final output of the HNFB^{-1} system is specified by the highest output obtained among all the T-conorm neurons, determining the class to which the input pattern belongs.

Results obtained with the HNFB^{-1} model, in different benchmark **classification** problems, are presented in Case Studies section.

HIERARCHICAL NEURO-FUZZY BPS MAMDANI

The Hierarchical Neuro-Fuzzy BSP Mamdani (HNFB-Mamdani), as HNFB^{-1}, was also developed to enhance the interpretability of the **hierarchical neuro-fuzzy systems**. However, since the HNFB^{-1} is dedicated to **classification** problems, a more general model was devised. The HNFB-Mamdani employs **Mamdani inference method** [Jang,1997] in the rules´ consequents, and can be applied in control systems, **pattern classification**, **forecasting**, and **rule extraction**.

HNFB-Mamdani Architecture

The HNFB-Mamdani architecture is formed by the interconnection of HNFB^{-1} cells in a binary tree structure and is divided into three basic modules: input partitioning structure; weighted connection from the binary structure leaf cells (d_i) to the T-conorm neurons (T_i); and the defuzzification process.

The first two modules are identical to the HNFB^{-1} architecture, except that each T-conorm neuron is associated with a fuzzy set M of the consequent. All leaf cells are connected to all T-conorm neurons. To each connection there is a weight associated, whose value is also establish by the Least Mean Squares algorithm.

The consequent of a fuzzy rule in the HNFB-Mamdani model is a fuzzy set represented by a triangular membership function. The total number of fuzzy sets associated with the output variable is specified by the user.

Defuzzification Method

The defuzzification process selected for the HNFB-Mamdani model is the weighted average of the maximum values. Figure 3 illustrates the defuzzification process for a model with three output fuzzy sets.

Figure 3. Defuzzification process

| Leaf Cells | T-Conorm Neurons | Fuzzy Sets | Defuzzification |

H
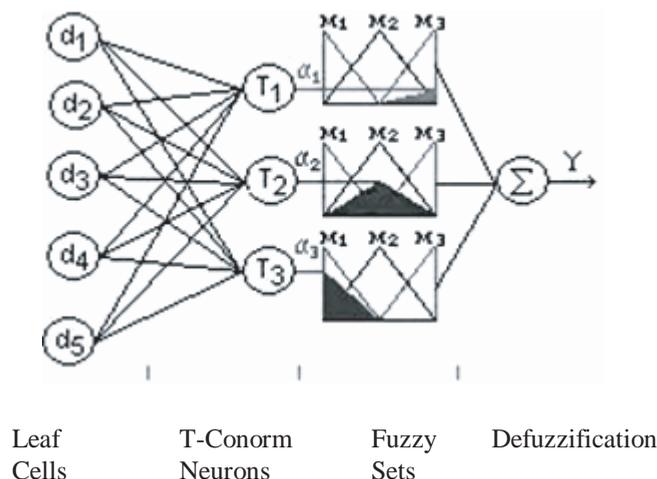
The output *y* is then calculated by Eq. (8).

$$y_j = \frac{\sum\limits_{i=1}^{n} \alpha_i^j * C_i}{\sum\limits_{i=1}^{n} \alpha_i^j}$$

(8)

where:

y_j: output of the HNFB-Mamdani for input pattern *j*;

α_i^j: output value of the *i*-th T-conorm neuron (T_i) for input pattern *j*;

C_i: value in the universe of discourse of the output variable where the M_i fuzzy set presents the maximum value;

* : product operator.

n : total number of fuzzy sets associated with the output variable.

CASE STUDIES

In order to evaluate the performance of supervised HNFB models, two benchmark classification databases and six load time series from utilities of the Brazilian electrical energy sector were selected.

Pattern Classification

Pattern classification aims to determine to which group of a pre-determined set an input pattern belong to. Two benchmark applications were selected among those most frequently employed in the area of machine learning. The results obtained with the proposed HNFB models were compared to the ones described in [Gonçalves,2006]. In order to generate the training and test sets, the total set of patterns was randomly divided into two equal parts. Each of these two sets was alternately used either as a training or as a test set. Table 1 below summarizes the average classification performance obtained with both test sets. The performance of the HNFB models is better than the other models, except for the HNFB-Mamdani case. Since HNFB-Mamdani is a general-purpose model, it tends to provide inferior results when compared to application-specific models, such as Inverted-HNFB and HNFB-Class. On the other hand, HNFB and HNFQ [Souza,2002a] are also general-purpose models but still

provide a superior performance than HNFB-Mamdani. This is due to the **Takagi-Sugeno inference method** used by those models, which is usually more accurate than the **Mamdani inference method** [Bezerra,2005]. The disadvantage of the **Takagi-Sugeno method** is its reduced interpretability.

Electric Load Forecasting

This experiment made use of data related to the monthly electric load of 6 utilities of the Brazilian electrical energy sector.

The results obtained with the HNFB models were compared with *Backpropagation* algorithm, statistical techniques, such as the *Holt-Winters* and *Box & Jenkins*,

Table 1. Comparison of the average classification performance

	Iris	Wine
NN	-	95.20 %
KNN	-	96.70%
FSS	-	92.80 %
BSS	-	94.80 %
MFS1	-	97.60%
MFS2	-	97.90 %
C4.5	94.00 %	-
FID3.1	96.00%	-
NEFCLASS	96.00 %	-
HNFB1	98.67 %	97.80%
HNFB2	98.67 %	97.80 %
HNFQ	98.67 %	98.88 %
HNFB-Inverted	98.67 %	99.44 %
HNFB-Class1	98.67 %	98.87 %
HNFB-Class2	97.33 %	98.88 %
HNFB-Mamdani	95,00 %	95.77%

where: NN=nearest-neighbor, KNN=k-nearest-neighbor, FSS=nearest-neighbor/forward sequential selection of feature), BSS=nearest-neighbor/backward sequential selection of feature), MFS=Multiple Feature Subsets, C4.5, FID3.1, NEFCLASS, HNFB1 (fixed selection), HNFB2 (adaptive selection), HNFQ, Inverted-HNFB, HNFB-Class1 (fixed selection) and HNFB-Class2 (adaptive selection). References to all these models are provided in [Gonçalves,2006].

Table 2. Monthly load prediction errors (MAPE) for different models

	HNFB-Mamdani	HNFB	Back Propagation	Box & Jenkins	Holt-Winters	RNB (Gaussian)	RNB (MCMC)
COPEL	1,77%	1,17 %	1.57%	1.63%	1.96%	1.45%	1.16%
CEMIG	1,39%	1,12 %	1.47%	1.67%	1.75%	1.29%	1.28%
LIGHT	2,41%	2,22 %	3.57%	4.02%	2.73%	1.44%	2.23%
FURNAS	3,08%	3,76 %	5.28%	5.43%	4.55%	1.33%	3.85%
CERJ	2,79%	1,35 %	3.16%	3.24%	2.69%	1.50%	1.33%
E.PAULO	1,42%	1,17 %	1.58%	2.23%	1.85%	0.79%	0.78%

and with Bayesian **Neural Nets** (BNN) [Bishop,1995], trained by Gaussian approximation and by the MCMC method. Table 2 below presents the performance results in terms of the "*Mean Absolute Percentage Error*".

It can be observed that the general performance of HNFB models is usually superior to the results provided by statistical methods. The results obtained with BNNs are generally better than with HNFB models. However, according to [Tito,1999], the training time with BNN was about 8 hours. This was a much longer period than the time required by the HNFB models to perform the same task, which was of the order of tens to hundreds of seconds, on similar equipment. Additionally, the data used in the HNFB models were not treated in terms of their seasonal aspects, nor were they made stationary as was the case of the BNN tested in [Tito,1999].

FUTURE TRENDS

As can be seen from the results presented, HNFB models provide very good performance in different applications. To improve the performance of the HNFB-Mamdani model, which provided the worst results among the supervised HNFB models, the model is being extended to allow the use of different types of output fuzzy sets (such as Gaussian, trapezoidal, etc.) and by adding an algorithm to optimize the total number of output fuzzy sets.

CONCLUSION

The objective of this article was to introduce a new class of **neuro-fuzzy models** which aims to improve the weak points of conventional **neuro-fuzzy systems**. The results obtained by the HNFB models showed that they yield a good performance as classifiers of database patterns or as time series forecasters. These models are able to create their own structure and allow the extraction of knowledge in the form of linguistic fuzzy rules.

REFERENCES

Abraham, A. (2005).Adaptation of Fuzzy Inference System Using Neural Learning, Fuzzy System Engineering: Theory and Practice, Springer-Verlag, Chapter3,pp.53-83.

Bezerra, R.A., Vellasco, M.M.B.R., Tanscheit, R. (2005).Hierarchical Neuro-Fuzzy BSP Mamdani System, 11th World Congress of International Fuzzy Systems Association,3,1321-1326.

Bishop, C.M. (1995).Neural Networks for Pattern Recognition, Clarendon Press.

Chrysanthou, Y. & Slater, M. (1992). Computing dynamic changes to BSP trees, EUROGRAPHICS '92,11(3),321-332.

H

Figueiredo, K.T., Vellasco, M.M.B.R. Pacheco, M.A.C. (2005a).Hierarchical Neuro-Fuzzy Models based on Reinforcement Learning for Intelligent Agents, Computational Intelligence and Bioinspired Systems,LNCS-3512,424-431.

Figueiredo, K., Santos, M., Vellasco, M.M.B.R., Pacheco, M.A.C. (2005b).Modified Reinforcement Learning-Hierarchical Neuro-Fuzzy Politree Model for Control of Autonomous Agents, International Journal of Simulation Systems, Science & Technology,6(10-11),4-13.

Gonçalves, L. B., Vellasco, M.M.B.R., Pacheco, M.A.C., Souza, F.J. (2006).Inverted Hierarchical Neuro-Fuzzy BSP System: A Novel Neuro-Fuzzy Model for Pattern Classification and Rule Extraction in Databases, IEEE Transactions on Systems, Man & Cybernetics,PartC,36(2),236-248.

Haykin, S. (1999). Neural Networks - A Comprehensive Foundation. Mcmillan College Publishing.

Jang, J.-S.R., Sun, C.-T., Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prenctice-Hall.

Nauck, D. & Kruse, R. (1997).A neuro_fuzzy method to learn fuzzy classification rules from data. Fuzzy Sets and Systems,88,277-288.

Nauck D. & Kruse, R. (1998).A Neuro-Fuzzy Approach to Obtain Interpretable Fuzzy Systems for Function Approximation, IEEE International Conference on Fuzzy Systems, 1106-1111.

Ross, T.J. (2004).Fuzzy Logic with Engineering Applications, John Wiley&Sons.

Souza, F.J., Vellasco, M.M.B.R., Pacheco, M.A.C. (2002a).Hierarchical neuro-fuzzy quadtree models, Fuzzy Sets and Systems 130(2),89-205.

Souza F.J., Vellasco, M.M.B.R., Pacheco, M.A.C. (2002b).Load Forecasting with The Hierarchical Neuro-Fuzzy Binary Space Partitioning Model, International Journal of Computers Systems and Signals 3(2),118-132.

Takagi, T. & Sugeno, M. (1985).Fuzzy identification of systems and its application to modelling and control, IEEE Trans. on Systems,Man and Cybernetics,15(1),116-132.

Tito, E., Zaverucha, G., Vellasco, M.M.B.R., Pacheco, M. (1999).Applying Bayesian Neural Networks to Electrical Load Forecasting, 6th Int. Conf. on Neural Information Processing.

Vellasco, M.M.B.R., Pacheco, M.A.C., Ribeiro-Neto, L.S., Souza, F.J.(2004).Electric Load Forecasting: Evaluating the Novel Hierarchical Neuro-Fuzzy BSP Model, International Journal of Electrical Power & Energy Systems, 26(2),131-142.

Vuorimaa, P.(1994).Fuzzy self-organizing map, Fuzzy Sets and Systems, 66(2),223-231.

Zhang,J. & Morris,A.J.(1995).Fuzzy neural networks for nonlinear systems modelling, IEE Proc.-Control Theory Appl.142(6),551-561.

KEY TERMS

Artificial Neural Networks: Composed of several units called neurons, connected through synaptic weights, which are iteratively adapted to achieve the desired response. Each neuron performs a weighted sum of its inputs, which is then passed through a nonlinear function that yields the output signal. ANNs have the ability to perform a non-linear mapping between their inputs and outputs, which is learned by a training algorithm.

Bayesian Neural Networks: Multi-layer neural networks that use training algorithms based on statistical Bayesian inference. BNNs offer a number of important advantages over the standard Backpropagation learning algorithm including: confidence intervals can be assigned to the predictions generated by a network; they allow the values of regularization coefficients to be selected using only training data; similarly, they allow different models to be compared using only the training data dealing with the issue of model complexity without the need to use cross validation.

Binary Space Partitioning: The space is successively divided in two regions, in a recursive way. This partitioning can be represented by a binary tree that illustrates the successive n-dimensional space sub-divisions in two convex subspaces. This process results in two new subspaces that can be later partitioned by the same method.

Fuzzy Logic: Can be used to translate, in mathematical terms, the imprecise information expressed by a set of linguistic IF-THEN rules. Fuzzy Logic studies the formal principles of approximate reasoning and is based on Fuzzy Set Theory. It deals with intrinsic imprecision, associated with the description of the properties of a phenomenon, and not with the imprecision associated with the measurement of the phenomenon itself. While classical logic is of a bivalent nature (true or false), fuzzy logic admits multivalence.

Machine Learning: Concerned with the design and development of algorithms and techniques that allow computers to "learn". The major focus of machine learning research is to automatically extract useful information from historical data, by computational and statistical methods.

Pattern Recognition: A sub-topic of machine learning, which aims to classify input patterns into a specific class of pre-defined groups. The classification is usually based on the availability of a set of patterns that have already been classified. Therefore, the resulting learning strategy is based on supervised learning.

Supervised Learning: A machine learning technique for creating a function from training data, which consist of pairs of input patterns as well as the desired outputs. Therefore, the learning process depends on the existance of a "teacher" that provides, to each input pattern, the real output value. The output of the function can be a continuous value (called regression), or a class label of the input object (called classification).

Hierarchical Neuro–Fuzzy Systems Part II

Marley Vellasco
PUC-Rio, Brazil

Marco Pacheco
PUC-Rio, Brazil

Karla Figueiredo
UERJ, Brazil

Flavio Souza
UERJ, Brazil

INTRODUCTION

This paper describes a new class of neuro-fuzzy models, called Reinforcement Learning Hierarchical Neuro-Fuzzy Systems (RL-HNF). These models employ the BSP (**Binary Space Partitioning**) and Politree partitioning of the input space [Chrysanthou,1992] and have been developed in order to bypass traditional drawbacks of **neuro-fuzzy systems**: the reduced number of allowed inputs and the poor capacity to create their own structure and rules (ANFIS [Jang,1997], NEFCLASS [Kruse,1995] and FSOM [Vuorimaa,1994]).

These new models, named Reinforcement Learning Hierarchical Neuro-Fuzzy BSP (RL-HNFB) and Reinforcement Learning Hierarchical Neuro-Fuzzy Politree (RL-HNFP), descend from the original HNFB that uses **Binary Space Partitioning** (see Hierarchical Neuro-Fuzzy Systems Part I). By using hierarchical partitioning, together with the **Reinforcement Learning** (RL) methodology, a new class of **Neuro-Fuzzy Systems** (SNF) was obtained, which executes, in addition to automatically learning its structure, the autonomous learning of the actions to be taken by an agent, dismissing a priori information (number of rules, fuzzy rules and sets) relative to the learning process. These characteristics represent an important differential when compared with existing intelligent **agents** learning systems, because in applications involving continuous environments and/or environments considered to be highly dimensional, the use of traditional **Reinforcement Learning** methods based on lookup tables (a table that stores value functions for a small or discrete state space) is no longer possible, since the state space becomes too large.

This second part of **hierarchical neuro-fuzzy systems** focus on the use of **reinforcement learning** process. The first part presented HNFB models based on supervised learning methods. The RL-HNFB and RL-HNFP models were evaluated in a benchmark **control application** and a simulated Khepera robot environment with multiple obstacles.

BACKGROUND

The model described in this paper was developed based on an analysis of the limitations in existing models and of the desirable characteristics for RL-based learning systems, particularly in applications involving continuous and/or high dimensional environments [Jouffe,1998][Sutton,1998][Barto,2003][**Satoh**,2006]. Thus, the *Reinforcement Learning Hierarchical Neuro-Fuzzy Systems* have been devised to overcome these basic limitations. Two different models of this class of **neuro-fuzzy systems** have been developed, based on **reinforcement learning** techniques.

HIERARCHICAL NEURO-FUZZY SYSTEMS

This section presents the new class of **neuro-fuzzy systems** that are based on hierarchical partitioning. As mentioned in the first part, two sub-sets of **hierarchical neuro-fuzzy systems** have been developed, according to the learning process used: supervised learning models (HNFB [Souza,2002][Vellasco,2004], HNFB[-1] [Gonçalves,2006], HNFB-Mamdani [Bezerra,2005]);

and **reinforcement learning** models (RL-HNFB [Figueiredo,2005a], RL-HNFP [Figueiredo,2005b]). The focus of this article is on the second sub-set of models. These models are described in the following sections.

REINFORCEMENT LEARNING HIERARCHICAL NEURO-FUZZY MODELS

The RL-HNFB and RL-HNFP models are composed of one or various standard cells, called RL-neuro-fuzzy-BSP (RL-NFB) and RL-neuro-fuzzy-Politree (RLNFP), respectively. The following sub-sections describe the basic cells, the hierarchical structures and the learning algorithm.

Reinforcement Learning Neuro-Fuzzy BSP and Politree Cells

An RL-NFB cell is a mini-neuro-fuzzy system that performs **binary partitioning** of a given space in accordance with ρ and μ membership functions. In the same way, an RL-NFP cell is a mini-neuro-fuzzy system that performs 2^n partitioning of a given input space, also using complementary membership functions in each input dimension. The RL-NFB and RL-NFP cells generate a precise (crisp) output after the defuzzification process [Figueiredo,2005a][Figueiredo,2005b].

The RL-NFB cell has only one input (x) associated with it. The RL-NFP cell receives all the inputs that are being considered in the problem. For illustration purpose, figure 1(a) depicts a cell with two inputs – x_1 and x_2 - (**Quadtree partitioning**), providing a simpler representation than the n-dimensional form of Politree. In figure 1(a) each partitioning is generated by the combination of two membership functions - ρ (*low*) and μ (*high*) of each input variable.

The consequents of the cell's poli-partitions may be of the *singleton* type or the output of a stage of a previous level. Although the *singleton* consequent is simple, this consequent is not previously known because each *singleton* consequent is associated with an action that has not been defined a priori. Each poli-partition has a set of possible actions (a_1, a_2, ... a_n), as shown in figure 1(a), and each action is associated with a Q-value function. The *Q*-value is defined as being the sum of the expected values of the rewards obtained

by the execution of action a in state s, in accordance with a policy π. For further details about RL theory, see [Sutton,1998].

The linguistic interpretation of the mapping implemented by the RL-NFP cell depicted in Figure 1(a) is given by the following set of rules:

$rule_1$: If $x_1 \in \rho_1$ and $x_2 \in \rho_2$ then $y = a_i$
$rule_2$: If $x_1 \in \rho_1$ and $x_2 \in \mu_2$ then $y = a_j$
$rule_3$: If $x_1 \in \mu_1$ and $x_2 \in \rho_2$ then $y = a_p$
$rule_4$: If $x_1 \in \mu_1$ and $x_2 \in \mu_2$ then $y = a_q$

where consequent a_i corresponds to one of the two possible consequents below:

a singleton (fuzzy singleton consequent*,* or zero-order Sugeno): the case where a_i=constant;
the output of a stage of a previous level: the case where $a_i = y_m$, where y_m represents the output of a generic cell 'm'.

RL-HNFB and RL-HNFP Architectures

RL-HNFB and RL-HNFP models can be created based on the interconnection of the basic cells. The cells form a hierarchical structure that results in the rules that compose the **agent**'s reasoning.

In the example of an architecture presented in figure 1(b), the poli-partitions *1*, *3*, *4*, …, *m-1* have not been subdivided, having as consequents of its rules the values a_1, a_3, a_4, …, a_{m-1}, respectively. On the other hand, poli-partitions 2 and m have been subdivided; so the consequents of its rules are the outputs (y_2 and y_m) of subsystems *2* and *m*, respectively. On its turn, these subsystems have, as consequent, the values a_{21}, a_{22}, ..., a_{2m}, and a_{m1}, a_{m2}, ..., a_{mm}, respectively. Each 'a_i' corresponds to a consequent of zero-order Sugeno (singleton), representing the action that will be identified (between the possible actions), through **reinforcement learning**, as being the most favorable for a certain state of the environment. It must be stressed that the definition of which partition must be subdivided or not is defined automatically by the learning algorithm.

The output of the system depicted in figure 1(b) (defuzzification) is given by equation (1). In these equations, α_i corresponds to the firing level of partition i and a_i is the singleton consequent of the rule associated with partition i.

H

Figure 1. (a) RL-NHP cell; (b) RL-HNFP architecture

(a)   *(b)*
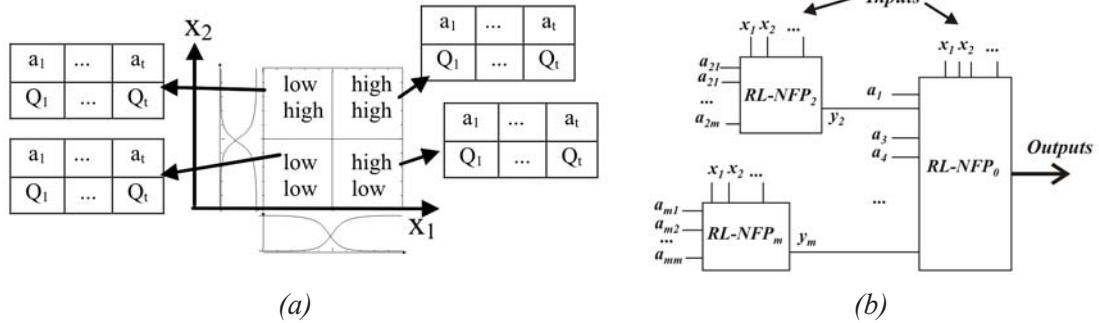
$$y = \alpha_1.a_1 + \alpha_2 \sum_{i=1}^{2^n} \alpha_{2i}.a_{2i} + \alpha_3.a_3 + \alpha_4.a_4 + \ldots + \alpha_m \sum_{i=1}^{2^n} \alpha_{mi}.a_{mi}$$

(1)

RL-HNFB and RL-HNFP Learning Algorithm

The learning process starts with the definition of the relevant inputs for the system/environment where the **agent** is and the sets of actions it may use in order to achieve its objectives. The **agent** must run many cycles to ensure learning in the system/environment where it is. A cycle is defined as the number of steps the agent takes in the environment, which extends from the point he is initiated to the target point.

The RL-HNFB and RL-HNFP models employ the same learning algorithm. Each partition chooses an action from its set of actions; the resultant action is calculated by the defuzzification process and represents the action that will be executed by the **agent**s' actuators. After the resultant action is carried out, the environment is read once again. This reading enables calculation of the environment reinforcement value that will be used to evaluate the action taken by the **agent**. The reinforcement is calculated for each partition of all active cells, by means of its participation in the resulting action. Thus, the environment reinforcement calculated by the evaluation function is backpropagated from the root-cell to the leaf-cells. Next, the Q-values associated to the actions that have contributed to the resulting action are updated, based on the SARSA

algorithm [Sutton,1998]. More details can be found in [Figueiredo,2005b].

The RL_HNFB and RL_HNFP models have been evaluated in different **control applications**. Two of these **control application** are presented in the next section.

CASE STUDIES

Cart-Centering

The *cart-centering problem* [Koza,1992] is generally used as a benchmark of the area of evolutionary programming, where the force that is applied to the car is of the "bang bang" type [Koza,1992]. This problem was used mainly for the purpose of evaluating how well the RL-HNFB and RL-HNFP models would adapt to changes in the input variable domain without having to undergo a new training phase.

The problem consists of parking, in the centre of a one-dimensional environment, a car with mass m that moves along this environment due to an applied force F. The input variables are the position (x) of the car, and its velocity (v). The objective is to park the car in position $x = 0$ with velocity $v = 0$. The equations of motion are (where the τ parameter represents the time unit):

$$x_{t+\tau} = x_t + \tau.v_t \qquad v_{t+\tau} = v_t + \tau.F_t / m$$

(2)

The global reinforcement is calculated by equation (3) below:

If (x>0 and v<0) or (x<0 and v>0)

$$R_{global} = k_1 e^{(-|distance\_objective|)} + k_2 e^{(|velocity|)} \qquad (3)$$

Else $R_{global} = 0$

The evaluation function increases as the car gets closer to the centre of the environment with velocity zero. The k_1 and k_2 coefficients are constants greater than 1 used for adapting the reinforcement values to the model's structure. The values used for time unit and mass were τ=0.02 and m=2.0.

The stopping criterion is achieved when the difference between the velocity and the position value in relation to the objective (x=0 and v=0) is smaller than 5% of the universe of discourse of the position and velocity inputs.

Table 1 shows the average of the results obtained in 5 experiments for each configuration. The columns position and velocity limits refer to the limits imposed to the (position and velocity) state variables during learning and testing. The actions used in these experiments are: F1={-150,-75,-50,-30,-20,-10,-5,0,5,10,20, 30,50,75,150}. The size of the structure column shows the average of the number of cells at the end of each experiment and the last column shows the average steps during the learning phase. The number of cycles was

Table 1. Results of the RL-HNFB and RL-HNFP models applied to the cart-centering problem

No.	Position Limits	Velocity Limits	Size of the Structure	Average steps learning phase
RL-HNFB$_1$	|10|	|10|	195 cells	424
RL-HNFB$_2$	|3|	|3|	340 cells	166
RL-HNFP$_1$	|10|	|10|	140 cells	221
RL-HNFP$_2$	|3|	|3|	251 cells	145

Table 2. Testing results of the proposed models applied to the cart-centering problem

Configuration	Initial Position		
	| 3|	|2|	|1|
	Average number of steps		
RL-HNFB$_1$	387	198	141
RL-HNFB$_2$	122	80	110
RL-NFHP$_1$	190	166	99
RL-NFHP$_2$	96	124	68

fixed at 1000. At each cycle, the car's starting points were *x*=-3 or *x*=3.

As can be observed from Table 1, the RL-HNFP structure is smaller because each cell receives both input variables, while in the case of the RL-HNFB model, a different input variable is applied at each level of the BSP tree.

Table 2 presents the results obtained for one of the 5 experiments carried out at each configuration shown in Table 1 when the car starts out at points (-2, -1, 1, 2) which were not used in the learning phase.

In the first configuration of each model, the results show that the broader the position and velocity limits are (in this case equal to |10|), the more difficult it is to learn. In these cases a small oscillation occurs around the central point. What actually happens is that the final velocity is very small but, after some time, it tends to move the car out of the convergence area, resulting in a peak of velocity in the opposite direction to correct the car's position. In the second configurations, fewer oscillations occur because the position and velocity limits were lowered to |3|.

Khepera Robot

The RL-HNFP model was also tested with a Khepera robot simulator [Figueiredo,2005b]. The model was tested in a squared environment where the **agent** moved from one of the corners to reach the diametric opposite corner. Nevertheless, he could not pass through the ambient center because of an obstacle.

The Khepera robot acquires ambient signs using 8 sensors grouped into 4: one ahead, one in each side and

one behind. Its actions are executed via 2 independent motors with power varying between -20 and 20, one in the right side and the other in the left side.

The RL-HNFP model was trained in one environment with a big central square obstacle, called environment I (Figure 2). It was tested in two environments: the same environment I (with a big central square obstacle) but with different initial positions; and another environment with four additional obstacles. These new multi-obstacle environment, (environment II), comprises: big central, small top-left, bottom-right and left obstacles (see Figure 3).

In all experiments using both environments the robot's objective was to reach position (5, 5). In figures 2 and 3 the white circles indicate the initial positions of the robot while the gray circles show their final positions.

Figure 2. Tests in environments I
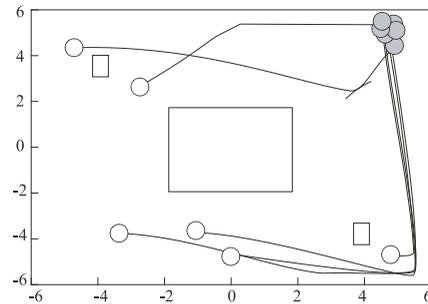
Figure 3. Tests in environments II

(a) (b) (c)
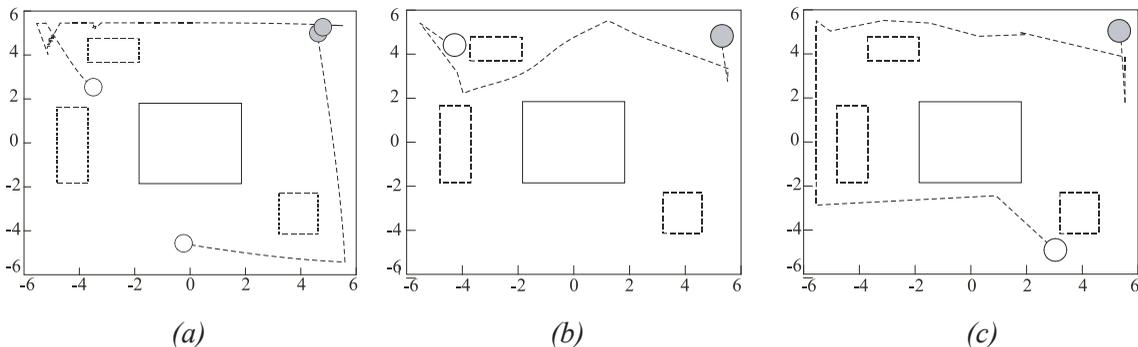
Figure 2 shows the experiments on the environment I. The results refer to tests executed with a structure trained from a variety of positions different from the ones used on the tests, demonstrating that the acquired knowledge was generalized and that the obtained results conform to the expectative.

Figures 3(a), 3(b) and 3(c) show experiments in environment II. The results described in these figures refer to tests executed in the environment II with the knowledge acquired in the environment I. Note that the three additional obstacles do not affect the results in these experiments.

As demonstrated from these figures, the results indicate the good performance of the model. This is even more important when one considers that the number of learning cycles was very small (only 400 cycles). The result presented in figure 3(c) stands out this fact. In this case, the **agent** does not take the shortest path to the goal point. On the contrary, it turns round the environment in some points until take the way to the goal point. To improve the robot's performance, the model should be executed for at least 2000 cycles, as already demonstrated from previous cases using this application [Figueiredo,2005b].

FUTURE TRENDS

Regarding the Reinforcement Learning Hierarchical neuro-fuzzy models, some improvements are also under development. To improve their performance it is intended to execute tests with *Eligibility Traces* [Sutton,1998]. This is a method that does not update only the current state function value, but also the function value of previous states inside a predefined limit.

Another proposal for improving model RL-NFHP is the use of the **WoLF** principle (Win or Learn Fast) [Bowling,2002] to modify the learning rate that adjusts the politics. The **WoLF** principle consists of learning quickly when it is losing and more slowly when it is winning. Also it is intended to evaluate these models using real robots.

The RL-HNFP model is also being modified to be used in a cooperative multi-agents environment. In this environment the learning process is accomplished by sharing the acquired knowledge among the existent agents.

CONCLUSION

The objective of this paper was to introduce a new class of **neuro-fuzzy models** which aims to improve the weak points of conventional **neuro-fuzzy systems**. The models RL-NFHB and RL-HNFP belong to this new class of **neuro-fuzzy systems** called Hierarchical Neuro-Fuzzy System.

The RL-NFHB and RL-HNFP models were able to create and expand the structure of rules without any prior knowledge (fuzzy rules or sets); extract knowledge from the agent's direct interaction with large and/or continuous environments (through **reinforcement learning**), in order to learn which actions are to be carried out; and produce interpretable fuzzy rules, which compose the agent's intelligence to achieve his goal(s). The agent was able to generalize its actions, showing adequate behaviour when the agent was in states whose actions had not been specifically learned. This capacity increases the agent's autonomy.

REFERENCES

Barto,A.G. and Mahadevan,S.(2003).Recent Advances in Hierarchical Reinforcement Learning,Volume13,Issue1-2, pp.41-77,ISSN:0924-6703.

Bezerra,R.A.,Vellasco,M.M.B.R.,Tanscheit,R.(2005). *Hierarchical Neuro-Fuzzy BSP Mamdani System,* 11th World Congress of International Fuzzy Systems Association (IFSA 2005),Vol.3,pp.1321-1326,Springer,July28-31,China.

Bowling.M and Veloso,M.(2002).Multiagent learning using a variable learning rate. *Artificial Intelligence*,136:215–250.

Bowling,M. and Veloso,M.(2001).Rational and convergent learning in stochastic games, In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence,pp.1021-1026,Seattle,WA.

Chrysanthou,Y., Slater,M.(1992).Computing dynamic changes to BSP trees, Computer Graphics Forum (EUROGRAPHICS'92Proceedings),11(3),321-332.

Gonçalves,L.B.,Vellasco,M.M.B.R.,Pacheco,M.A.C., Souza,F.J.(2006).*Inverted Hierarchical Neuro-Fuzzy*

BSP System: A Novel Neuro-Fuzzy Model for Pattern Classification and Rule Extraction in Databases, IEEE Transactions on Systems, Man & Cybernetics, Part C: Applications and Review, Vol.36,No.2,pp.236-248.

Figueiredo,K, Vellasco,M.M.B.R., Pacheco,M.A.C.(2005a).*Hierarchical Neuro-Fuzzy Models based on Reinforcement Learning for Intelligent Agents,* Lecture Notes in Computer Science - Computational Intelligence and Bioinspired Systems, Volume 3512,Springer,pp.424-431.

Figueiredo, K., Santos, M., Vellasco,M.M.B.R., Pacheco,M.A.C.(2005b).*Modified Reinforcement Learning-Hierarchical Neuro-Fuzzy Politree Model for Control of Autonomous Agents,* special issue on Soft Computing for Modeling and Simulation" of the International Journal of Simulation Systems, Science & Technology, UK,Vol. 6, Nos.10 and 11,pp.4-13.

Jang, J.-S.R., Sun, C.-T., Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prenctice-Hall.

Jouffe,L.(1998).Fuzzy Inference System Learning by Reinforcement Methods, IEEE Trans. on SMCs-C 28/3,pp.338-355.

Koza,J.R.(1992).Genetic Programming: On the programming of computers by means of natural selection, Cambridge,MA, MIT Press,(1992).

Kruse,R. and Nauck,D.(1995).NEFCLASS-A neuro-fuzzy approach for the classification of data, *Proc. of ACM Symposium on Applied Computing,* Nashville,pp. 461-465.

Satoh,H.(2006).A State Space Compression Method Based on Multivariate Analysis for Reinforcement Learning in High-Dimensional Continuous State Spaces, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E89-A(8): pp.2181-2191.

Sutton,R.S. and Barto,A.G.(1998).Reinforcement Learning: An Introduction,MIT Press.

Souza,F.J.,Vellasco,M.M.B.R.,Pacheco,M.A.C.(2002). Load Forecasting with The Hierarchical Neuro-Fuzzy Binary Space Partitioning Model, International Journal of Computers Systems and Signals, South Africa Vol.3,No.2, pp.118-132.

Vellasco,M.M.B.R., Pacheco,M.A.C., Ribeiro Neto,L.S., Souza,F.J.(2004).Electric Load Forecasting: Evaluating the Novel Hierarchical Neuro-Fuzzy BSP Model, International Journal of Electrical Power & Energy Systems, Vol.26,No.2,pp.131-142, Elsevier Science Ltd, February.

Vuorimaa,P.(1994).Fuzzy self-organizing map, *Fuzzy Sets and Systems,*vol.66,no. 2,pp.223-231.

KEY TERMS

Binary Space Partitioning: In this type of partitioning, the space is successively divided in two regions, in a recursive way. This partitioning can be represented by a binary tree that illustrates the successive n-dimensional space sub-divisions in two convex subspaces. The construction of this partitioning tree (BSP tree) is a process in which a subspace is divided by a hyper-plan parallel to the co-ordinates axes. This process results in two new subspaces that can be later partitioned by the same method.

Fuzzy Inference Systems: Fuzzy inference is the process of mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis.

Machine Learning: Concerned with the design and development of algorithms and techniques that allow computers to "learn". The major focus of machine learning research is to automatically extract useful information from historical data, by computational and statistical methods.

Politree Partitioning: The Politree partitioning was inspired by the quadtree structure, which has been widely used in the area of images manipulation and compression. In the politree partitioning the subdivision of the n-dimensional space is accomplished by $m=2^n$ subdivision. The Politree partitioning can be represented by a tree structure where each node is subdivided in m leafs (Politree partitioning).

Quadtree Partitioning: In this type of partitioning, the space is successively divided in four regions, in a recursive way. This partitioning can be represented

by a quaternary tree that illustrates the successive n-dimensional space sub-divisions in four convex sub-spaces. The construction of this partitioning tree (Quad tree) is a process in which a subspace is divided by a two hyper-plan parallel to the co-ordinates axes. This process results in four new subspaces that can be later partitioned by the same method. The limitation of the Quadtree partitioning (fixed or adaptive) is in the fact that it works only in two-dimensional spaces.

Reinforcement Learning: A sub-area of machine learning concerned with how an *agent* ought to take *actions* in an *environment* so as to maximize some notion of long-term *reward*. Reinforcement learning algorithms attempt to find a *policy* that maps *states* of

the world to the actions the agent ought to take in those states. Differently from supervised learning, in this case there is no target value for each input pattern, only a reward based of how good or bad was the action taken by the agent in the existant environment.

Sarsa: It is a variation of the Q-learning (Reinforcement Learning) algorithm based on model-free action policy estimation. SARSA admits that the actions are chosen randomly with a predefined probability.

WoLF: ("Win or Learn Fast") is a method by [Bowling,2002] for changing the learning rate to encourage convergence in a multi-agents reinforcement learning scenario.

Hierarchical Reinforcement Learning

Carlos Diuk
Rutgers University, USA

Michael Littman
Rutgers University, USA

INTRODUCTION

Reinforcement learning (RL) deals with the problem of an agent that has to learn how to behave to maximize its utility by its interactions with an environment (Sutton & Barto, 1998; Kaelbling, Littman & Moore, 1996). Reinforcement learning problems are usually formalized as Markov Decision Processes (MDP), which consist of a finite set of states and a finite number of possible actions that the agent can perform. At any given point in time, the agent is in a certain state and picks an action. It can then observe the new state this action leads to, and receives a reward signal. The goal of the agent is to maximize its long-term reward.

In this standard formalization, no particular structure or relationship between states is assumed. However, learning in environments with extremely large state spaces is infeasible without some form of generalization. Exploiting the underlying structure of a problem can effect generalization and has long been recognized as an important aspect in representing sequential decision tasks (Boutilier et al., 1999).

Hierarchical Reinforcement Learning is the subfield of RL that deals with the discovery and/or exploitation of this underlying structure. Two main ideas come into play in hierarchical RL. The first one is to break a task into a hierarchy of smaller subtasks, each of which can be learned faster and easier than the whole problem. Subtasks can also be performed multiple times in the course of achieving the larger task, reusing accumulated knowledge and skills. The second idea is to use state abstraction within subtasks: not every task needs to be concerned with every aspect of the state space, so some states can actually be *abstracted away* and treated as the same for the purpose of the given subtask.

BACKGROUND

In this section, we will introduce the MDP formalism, where most of the research in standard RL has been done. We will then mention the two main approaches used for learning MDPs: model-based and model-free RL. Finally, we will introduce two formalisms that extend MDPs and are widely used in the Hierarchical RL field: semi-Markov Decision Processes (SMDPs) and Factored MDPs.

Markov Decision Processes (MDPs)

A Markov Decision Process consists of:

- a set of states S
- a set of actions A
- a transition probability function: *Pr(s' | s, a),* representing the probability of the environment transitioning to state *s'* when the agent performs action *a* from state *s*. It is sometimes notated *T(s, a, s')*.
- a reward function: *E[r | s, a]*, representing the expected immediate reward obtained by taking action *a* from state *s*.
- a discount factor $\gamma \in (0, 1]$, that downweights future rewards and whose precise role will be clearer in the following equations.

A *deterministic policy* $\pi{:}S \rightarrow A$ is a function that determines, for each state, what action to take. For any given policy π, we can define a *value function* V^{π}, representing the *expected infinite-horizon discounted return* to be obtained from following such a policy starting at state *s*:

$$V^{\pi}(s) = E[r_0 + \gamma\, r_1 + \gamma^2\, r_2 + \gamma^3\, r_3 + \dots].$$

Bellman (1957) provides a recursive way of determining the value function when the reward and transition probabilities of an MDP are known, called the *Bellman equation*:

$$V^\pi(s) = R(s, \pi(s)) + \gamma \Sigma_{s'} \ T(s, \pi(s), s') \ V\pi(s'),$$

commonly rewritten as an *action-value function* or *Q-function*:

$$Q\pi(s,a) = R(s, a) + \gamma \Sigma_{s'} \ T(s, a, s') \ V\pi(s').$$

An *optimal policy* $\pi^*(s)$ is a policy that returns the action *a* that maximizes the value function:

$$\pi^*(s) = argmax_a \ Q^*(s,a)$$

States can be represented as a set of *state variables or factors*, representing different features of the environment: $s = <f_1, f_2, f_3, \dots, f_n>$.

Learning in Markov Decision Processes (MDPs)

The reinforcement-learning problem consists of determining or approximating an optimal policy through repeated interactions with the environment (*i.e.*, based on a sample of *experiences* of the form <*state – action – next state – reward*>).

There are three main approaches to learning such an optimal or near-optimal policy:

- **Policy-search methods:** learn a policy directly via evaluation in the environment.
- **Model-free (or direct) methods:** learn the policy by directly approximating the *Q* function with updates from direct experience.
- **Model-based (or indirect) methods:** first learn the transition probability and reward functions, and use those to compute the *Q* function by means of , for example, the *Bellman equations*.

Model-free algorithms are sometimes referred to as the *Q-learning family* of algorithms. See Sutton (1988) or Watkins (1989) for the first best-known examples. It is known that model-free methods make inefficient use of experience, but they do not require expensive

computation to obtain the *Q* function and the corresponding optimal policy.

Model-based methods make more efficient use of experience, and thus require less data, but they involve an extra *planning* step to compute the value function, which can be computationally expensive. Some well-known algorithms can be found in the literature (Sutton, 1990; Moore & Atkeson, 1993; Kearns & Singh, 1998; and Brafman & Tennenholtz, 2002).

Algorithms for reinforcement learning in MDP environments suffers from what is known as the *curse of dimensionality*: an exponential explosion in the total number of states as a function of the number of *state variables*. To cope with this problem, *hierarchical methods* try to break down the intractable state space into smaller pieces, which can be learned independently and reused as needed. To achieve this goal, changes need to be introduced to the standard MDP formalism. In the introduction we mentioned the two main ideas behind hierarchical RL: task decomposition and state abstraction. Task decomposition implies that the agent will not only be performing single-step actions, but also full subtasks which can be extended in time. Semi-Markov Decision Processes (SMDPs) will let us represent these extended actions. State abstraction means that, in certain contexts, certain aspects of the state space will be ignored, and states will be grouped together. Factored-state representations is one way of dealing with this. The following section introduces these two common formalisms used in the HRL literature.

Beyond MDPs: SMDPs and Factored-State Representations

We'll consider the limitations of the standard MDP formalism by means of an illustrating example. Imagine an agent whose task is to exit a multi-storied office building. The starting position of the agent is a certain office in a certain floor, and the goal is to reach the front door at ground level. To complete the task, the agent has to first exit the room, find its way through the hallways to the elevator, take the elevator to the ground floor, and finally find its way from the elevator to the exit. We would like to be able to reason in terms of subtasks (e.g., *"exit room"*, *"go to elevator"*, *"go to floor X"*, etc.), each of them of different durations and levels of abstraction, each encompassing a series of

H

lower-level or *primitive* actions. Each of these subtasks is also concerned with only certain aspects of the full state space: while the agent is inside the room, and the current task is to exit it, the floor the elevator is on, or whether the front door of the building is open or closed, is irrelevant. However, these features will become crucial later as the agent's subtask changes.

Under the MDP formalization, time is represented as a discrete step of unitary and constant duration. This formulation does not allow the representation of temporally extended actions of varying durations, amenable to represent the kind of higher-level actions identified in the example. The formalism of semi-Markov Decision Processes (SMDPs) enables this representation (Puterman, 1994). In SMDPs, the transition function is altered to represent the probability that action *a* from state *s* will lead to next state *s'* after *t* timesteps:

$Pr(s', t \mid s, a)$

The corresponding value function is now:

$V^{\pi}(s) = R(s, \pi(s)) + \Sigma_{s' \in S} \gamma^t Pr(s', t \mid s, a) V\pi(s')$

SMDPs also enable the representation of continuous time. For dynamic programming algorithms for solving SMDPs, see Puterman (1994) and Mahadevan et al., (1997).

Factored-state MDPs deal with the fact that certain aspects of the state space are irrelevant for certain actions. In factored-state MDPs, state variables are decomposed into independently specified components, and transition probabilities are defined as a product of factor probabilities. A common way of representing independence relations between state variables is through *Dynamic Bayes Networks* (DBNs). As an example, imagine that the state is represented by four state variables: $s = \langle f_1, f_2, f_3, f_4 \rangle$, and we know that for action *a* the value of variable f_1 in the next state only depends on the prior values of f_1 and f_4, f_2 depends on f_2 and f_3, and the others only depend on their own prior value. This transition probability in a Factored MDP would be represented as:

$Pr(s' \mid s, a) = Pr(f_1' \mid f_1 f_4, a) \, Pr(f_2' \mid f_2 f_3, a) \, Pr(f_3' \mid f_3, a) \, Pr(f_4' \mid f_4, a)$

For learning algorithms in factored-state MDPs, see Kearns & Koller (1999) and Guestrin et al. (2002).

HIERARCHICAL REINFORCEMENT-LEARNING METHODS

Different approaches and goals can be identified within the hierarchical reinforcement-learning subfield. Some algorithms are concerned with learning a hierarchical view of either the environment or the task at hand, while others are just concerned with exploiting this knowledge when provided as input. Some techniques try to learn or exploit temporally extended actions, abstracting together a set of actions that lead to the completion of a subtask or subgoal. Other methods try to abstract together different states, treating them as if they were equal from the point of view of the learning problem.

We will briefly review a set of algorithms that use some combination of these approaches. We will also identify which of these methods are based on the model-free learning paradigm as opposed to those that try to construct a model of the environment.

Options: Learning Temporally Extended Actions in the SMDP Framework

Options make use of the SMDP framework to allow the agent to group together a series of actions (an option's policy*)* that lead to a certain state or set of states identified as subgoals. For each option, a set of valid start states is also identified, where the agent can decide whether to perform a single-step primitive action, or to make use of the option. We can think of options as pre-stored policies for performing abstract subtasks.

A learning algorithm for options is described by Sutton, Precup & Singh (1999) and belongs to the model-free Q-learning family. In its current formulation, the options framework allows for two-level hierarchies of tasks, although they could potentially be generalized to multiple levels. End states (*i.e.*, subgoals) are given as input to the algorithm. There is work devoted to discovering these subgoals and constructing useful options from them (Şimşek et al., 2005; and Jong & Stone, 2005).

While *options* have been shown to improve the learning time of model-free algorithms, it is not clear that there is an advantage in terms of learning time over model-based methods. As any model-free method, though, they do not suffer from the computational cost involved in the planning step. It is still an open question whether options can be generalized to multiple-level hierarchies, and most of the work is empirical, with no theoretical bounds.

MaxQ: Combining a Hierarchical Task Decomposition with State Abstraction

MaxQ is also a model-free algorithm in the Q-learning family. It receives as input a multi-level hierarchical task decomposition, which decomposes the full underlying MDP into an additive combination of smaller MDPs. Within each task, abstraction is used so that state variables that are irrelevant for the task are ignored (Dieterich, 2000).

The main drawback of *MaxQ* is that the hierarchy and abstraction have to be provided as input, and in it's model-free form it misses opportunities for faster learning.

DSHP: Model-Based Hierarchical Decomposition for Efficient Learning and Planning

Deterministic Sample-Based Hierarchical Planning (DSHP) combines factored-state MDP representations, a *MaxQ* hierarchical task decomposition, and model-based learning to achieve provably efficient learning and planning in deterministic domains (Diuk, Strehl & Littman, 2006).

While, as a model-based algorithm, *DSHP* allows for faster learning and planning, it still suffers from the problem that the hierarchy and abstraction have to be provided as input.

HEXQ: Discovering Hierarchy

As opposed to *MaxQ, DSHP*, or other methods that receive the hierarchical task decomposition as input, *HEXQ* tries to automatically discover it. *HEXQ* analyses traces of experience and identifies regions of the MDP with repeated characteristics. It uses this experience to build temporal and state abstractions, constructing a hierarchy of smaller interlinked MDPs. *HEXQ* is model-free and based on Q-learning (Hengst, 2002).

HEXQ shows a promising method for discovering abstractions and hierarchies, but still suffers from a lack of any theoretical bounds or proofs. All the work using *HEXQ* has been empirical, and it's general power still remains an open question.

HAM-PHAM: Restricting the Class of Possible Policies

Hierarchies of Abstract Machines (HAMs) also make use of the SMDP formalism. The main idea is to restrict the class of possible policies by means of small nondeterministic finite-state machines, which constrain the sequences of actions that are allowed. Elements in *HAMs* can be thought of as small programs, which at certain points can decide to make calls to other lower-level programs (Parr & Russell, 1997; andParr, 1998). See also *Programmable HAMs (PHAMs)*, an extension by Andre & Russell (2000).

HAM provides an interesting approach to make learning and planning easier, but has also only been shown to work better in certain empirical examples.

FUTURE TRENDS

We expect to see most of the new work in the field of Hierarchical Reinforcement Learning tackling two areas: hierarchy and abstraction discovery, and transfer learning. We believe the main open question is how structure can be learned from experience, and once learned be applied to tasks and problems different from the original one.

There is also promising but still little theoretical work currently being produced in the area, work that could prove the general power of different methods. Most of the work is empirical and only shown to work through experiments in small domains.

CONCLUSION

The goal of hierarchical reinforcement learning is to combat the "curse of dimensionality", the main obstacle in achieving scalable RL that can be applied to real-life problems, by means of hierarchical task decompositions and state abstraction. This active area of research has

H

achieved mixed results, with algorithms and frameworks focusing on just one or two combinations of the different aspects of the problem. A single approach that can deal with structure discovery and its use, with both temporal and state abstraction, and that can provably learn and plan in polynomial time is still the main item in the research agenda of the field.

REFERENCES

Andre, D. & Russell. S. J. (2000). Programmable reinforcement learning agents. *Advances in Neural Information Processing Systems (NIPS)*.

Barto, A.G. & Mahadevan, S. (2003). Recent Advances in Hierarchical Reinforcement Learning. *Special Issue on Reinforcement Learning, Discrete Event Systems Journal.* (13) 41-77.

Bellman, R. (1957). *Dynamic Programming.* Princeton University Press.

Boutilier, C.; Dean, T.; & Hanks, S. (1999) Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research.* (11) 1-94.

Brafman, R. & Tennenholtz, M. (2002). R-MAX – a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*.

Dietterich, T.G. (2000). Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research* (13) 227–303.

Diuk, C.; Strehl, A. & Littman, M.L. (2006). A Hierarchical Approach to Efficient Reinforcement Learning in Deterministic Domains. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Guestrin, C.; Patrascu, R.; & Schuurmans, D. (2002). Algorithmdirected exploration for model-based reinforcement learning in factored MDPs. *Proceedings of the International Conference on Machine Learning*, 235–242.

Hengst, B. (2002). Discovering hierarchy in reinforcement learning with hexq. *Proceedings of the 19th International Conference on Machine Learning*.

Jong, N & Stone, P. (2005) State Abstraction Discovery from Irrelevant State Variables. *Proceedings of the 19th International Joint Conference on Artificial Intelligence.*

Kaelbling, L. P.; Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*. (4) 237-285.

Kearns, M. & Singh, S. (1998). Near-Optimal Reinforcement Learning in Polynomial Time. *Proceedings of the 15th International Conference on Machine Learning*.

Kearns, M. J., & Koller, D. (1999). Efficient reinforcement learning in factored MDPs. *In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, 740–747.

Mahadevan, S., Marchalleck, N., Das, T. & Gosavi, A. (1997). Self-improving factory simulation using continuous-time average-reward reinforcement learning. *Proceedings of the 14th International Conference on Machine Learning.*

Moore, A. & Atkeson, Ch. (1993). Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*.

Parr, R. & Russell, S. (1997). Reinforcement learning with hierarchies of machines. *Proceedings of Advances in Neural Information Processing Systems 10*.

Parr, R. (1998). Hierarchical Control and learning for Markov decision processes. *PhD thesis, University of California at Berkeley*.

Puterman, M. L. (1994). *Markov Decision Problems*. Wiley, New York.

Şimşek, Ö, Wolfe, A.P. & Barto, A. (2005). Identifying useful subgoals in reinforcement learning by local graph partitioning. *Proceedings of the 22nd International Conference on Machine Learning*

Sutton, R. S. (1988). Learning to predict by the method of temporal differences. *Machine Learning*.

Sutton, R. S. (1990). Integrated architectures for learning, planning and reacting based on approximating dynamic programming. *Proceedings of the 7th International Conference on Machine Learning*.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. The MIT Press.

Sutton, R.; Precup, D. & Singh, S. (1999) Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*

Watkins, C. (1989). Learning from Delayed Rewards. *PhD Thesis.*

KEY TERMS

Factored-State Markov Decision Process: An extension to the MDP formalism used in Hierarchical RL where the transition probability is defined in terms of factors, allowing the representation to ignore certain state variables under certain contexts.

Hierarchical Reinforcement Learning: A subfield of reinforcement learning concerned with the discovery and use of task decomposition, hierarchical control, temporal and state abstraction (Barto & Mahadevan, 2003).

Hierarchical Task Decomposition: A decomposition of a task into a hierarchy of smaller subtasks.

Markov Decision Process: The most common formalism for environments used in reinforcement learning, where the problem is described in terms of a finite set of states, a finite set of actions, transition probabilities between states, a reward signal and a discount factor.

Reinforcement Learning: The problem faced by an agent that learns to a utility measure behavior from its interaction with the environment.

Semi-Markov Decision Process: An extension to the MDP formalism that deals with temporally extended actions and/or continuous time.

State-Space Generalization: The technique of grouping together states in the underlying MDP and treating them as equivalent for certain purposes.

High Level Design Approach for FPGA Implementation of ANNs

Nouma Izeboudjen
Center de Développement des Technologies Avancées (CDTA), Algérie

Ahcene Farah
Ajman University, UAE

Hamid Bessalah
Center de Développement des Technologies Avancées (CDTA), Algérie

Ahmed Bouridene
Queens University of Belfast, UK

Nassim Chikhi
Center de Développement des Technologies Avancées (CDTA), Algérie

INTRODUCTION

Artificial neural networks (ANNs) are systems which are derived from the field of neuroscience and are characterized by intensive arithmetic operations. These networks display interesting features such as parallelism, classification, optimization, adaptation, generalization and associative memories. Since the McCulloch and Pitts pioneering work (McCulloch, W.S., & Pitts, W. (1943), there has been much discussion on the topic of ANNs implementation, and a huge diversity of ANNs has been designed (C. Lindsey & T. Lindblad, 1994). The benefits of using such implementations is well discussed in a paper by R. Lippmann (Richard P. Lipmann, 1984): "The great interest of building neural networks remains in the high speed processing that can be achieved through massively parallel implementation". In another paper Clark S. Lindsey (C.S Lindsey, Th. Lindbald, 1995) posed a real dilemma of hardware implementation: "Built a general, but probably expensive system that can be reprogrammed for several kinds of tasks like CNAPS for example? Or build a specialized chip to do one thing but very quickly, like the IBM ZISC Processor". To overcome this dilemma, most researchers agree that an ideal solution should relay the performances obtained using specific hardware implementation and the flexibility allowed by software tools and general purpose chips.

Since their commercial introduction in the mid-1980's, and due to the advances in the development of both of the microelectronic technology and the specific CAD tools, FPGAs devices have progressed in an evolutionary and revolutionary way. The evolution process has allowed faster and bigger FPGAs, better CAD tools and better technical support. The revolution process concerns the introduction of high performances multipliers, Microprocessors and DSP functions. This has a direct incidence to FPGA implementation of ANNs and a lot of research has been carried to investigate the use of FPGAs in ANNs implementation (Amos R. Omandi & Jagath C. rajapakse, 2006).

Another attractive key feature of FPGAs is their flexibility, which can be obtained at different levels: exploitation of the programmability of FPGA, dynamic reconfiguration or run time reconfiguration (RTR), (Xilinx XAPP290, 2004) and the application of the **design for reuse** concept (Keating, Michael; Bricaud, Pierre, 2002).

However, a big disadvantage of FPGAs is the low level hardware oriented programming model needed to fully exploit the FPGA's potential performances.

High level based VHDL synthesis tools have been proposed to bridge the gap between the high level application requirements and the low level FPGA hardware but these tools are not algorithmic or application specific. Thus, special concepts need to be developed for automatic ANN implementation before using synthesis tools.

In this paper, we present a high level design methodology for ANN implementation that attempts to build a

bridge between the synthesis tool and the ANN design requirements. This method offers a high flexibility in the design while achieving speed/area performances constraints. The three implementation figures of the ANN based back propagation algorithm are considered. These are the off-type implementation, the on-chip global implementation and the dynamic reconfiguration choices of the ANN.

To achieve our goal, a design for reuse strategy has been applied. To validate our approach, three case studies are considered using the Virtex-II and Virtex-4 FPGA devices. A comparative study is done and new conclusions are given.

BACKGROUND

In this section, theoretical presentation of the **multilayer perceptron** (MLP) based back propagation algorithm is given. Then, discussion of the most related works to the topics of high level design methodology and ANNs frameworks are given.

Theoretical Background of the Back Propagation Algorithm

The back propagation is one of the well known algorithms that are used to train the MLP ANN network in a supervised mode. The MLP is executed in three phases: the f*eed forward phase*, the *error calculation phase* and the *synaptic weight updating* phase (Freeman, J. A. and Skapura, D. M, 1991).

In the *feed forward phase*, a pattern x_i is applied to the input layer and the resulting signal is forward propagated through the network until the final outputs have been calculated; for each *i* (index of neuron) and j (index of layer)

$$\mu_j^{[i]} = \sum_{i=1}^{n_0} W_j^{[i]} x_i \tag{1}$$

$$o_j^i = f(x_j) = \frac{1}{1 + \exp(-\mu_j)} \tag{2}$$

where, μ_j^i is the weighted sum of the synaptic weights and o_j^i is the output of the sigmoid activation function.

The e*rror calculation step*, computes the local error, δ for each layer starting from output back to input:

$$\delta_i^L = f'(u_i^L)(d_i - y_i) \tag{3}$$

$$\delta_j^{l-1} = f'(u_j^{l-1}) \sum_{i=1}^{N_l} w_{ij} \delta_i^l \quad 1 \le i \le N_l \quad , \quad 1 \le l \le L \tag{4}$$

where, d_i is the desired output f' the derivative function of f

The *Weight update step* computes the weights updates according to:

$$w_{ij}^l(t+1) = w_{ij}^l(t) + \Delta w_{ij}^l(t) \tag{5}$$

$$\Delta w_{ij}^l(t) = \eta \delta_i^l y_j^{l-1} \tag{6}$$

where, η is the learning factor, Δ*w* the variation of weights and *l*, the indices of the layers.

Background on ANN Frameworks

The most related works to ANNs frameworks are presented by (F. Schurmann & all, 2002), (M. Diepenhorst & all, 1999), and (J. Zhu & all, 1999).

In the other hand, and with the increasing complexity of FPGAs **circuits**, Core-based synthesis methodology is proposed as a new trend for efficient hardware implementation of FPGAs. In these tools a library of pre-designed IPs "Intellectual Property" cores are proposed. An example can be found in (Xilinx Core Generator reference) and (Opencores reference).

In the core based design methodology, efficient reuse is derived from the parameterized design with VHDL and its many flexible constructs and characteristics (i.e. abstraction, encapsulation, inheritance and reuse through attributes, package, procedures and functions). Beside this, the reuse concept is well suited for high regular and repetitive structures such as neural networks. However although all these advantages, seldom attention has been done to apply **design for reuse** for ANNs.

In this context our paper presents a new high level design methodology based upon the use of the **design for reuse** concept for ANNs.

In order to achieve this goal, the design must fulfill the following requirements (Keating, Michael; Bricaud, Pierre, 2002):

• The design must be block-based

- The design must be reconfigurable to meet the requirement of many different applications.
- The design must use standard interfaces.
- The code must be synthesizable at the RTL level.
- The design must be verified.
- The design must have robust scripts and must be well documented.

PRESENTATION OF THE PROPOSED DESIGN APPROACH

The proposed design approach is shown in Fig.1 as a process of flow. In this figure, the methodology used is based on a **top down design** approach in which the designer/user is guided step by step in the design process of the ANN.

First, the user is asked to select the dimension of the network. The next step involves selection of ANN implementation choices; these are the off chip

implementation, the global on chip implementation and implementation using run time reconfiguration (RTR). Thus a **Core** is generated for each type of implementation.

At this level, the user/designer can fix the parameters of the network, i.e. the number of neurons in each layer, synaptic values, multiplier type, data representation and precision. At the low level all the IP Cores that construct the neuron are generated automatically from the library of the synthesis tool which is in our case MENTOR GRAPHICS (Mentor Graphics user guide reference), and which also integrates the Xilinx IP **Core** Generator. In addition, for each IP **Core,** a graphical interface is generated to fix its parameters. Thus, the user/designer can change the network performances architecture by changing the IP cores that are stoked in the library. Then a VHDL code at the register transfer level (RTL) is generated for synthesis. Before, functional simulation is required. The result is a file netlist ready for place and rout followed by final FPGA prototyping on a board. Documentation is available at each level of the

Figure 1. The proposed design methodology
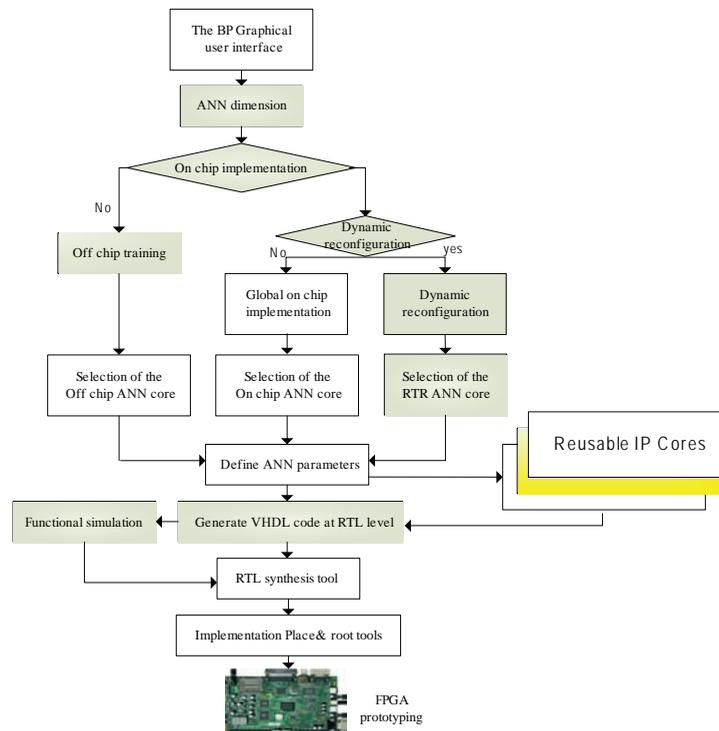
design process and the code is well commented. Thus, the **design for reuse** requirements is applied through the design process. In what follow, presentation of each implementation type is given.

The Feed Forward Off-Chip Implementation

Fig. 2 shows a top view of the feed forward core which is composed of a data path **module** and a control **module**. At the top level these two **modules** are represented by black boxes and only the neural network inputs and outputs signals are shown to the user/designer.

By clicking inside the boxes, we can get access to the network architecture which is composed of three layers represented by black boxes as shown in Fig. 3 (left side). By clicking inside each box, we can get access to the layer architecture which is composed of black boxes representing the neurons as shown in Fig. 3 (right side); and by clicking inside each neuron's box we can get access to the neuron hardware architecture as shown in Fig 4.

Each neuron implements the accumulated weight sum of equation (1) and the activation function of

equation (2). As shown in Fig. 4, the hardware model of the neuron is mainly based on a:

- Memory **circuit** where the final values of the synaptic weights are stocked,
- A multiply **circuit** (MULT) which computes the product of the stored synaptic weights with inputs data
- An accumulator circuit (ACUM) which computes the sum of the above products
- A **circuit** that approximates the activation function (example linear function or sigmoid function)
- A multiplexer **circuit** (MUX) in the case of serial transfer between inputs in the same neuron

The neural network architecture has the following properties:

- Computation between layers is done serially
- For the same layer, neurons are computed in parallel
- For the same neuron, only one multiplier and one accumulator (MULT +ACUM=MAC) are used to compute the product sum.

Figure 2.The feed forward core module using the mentor graphics design tool
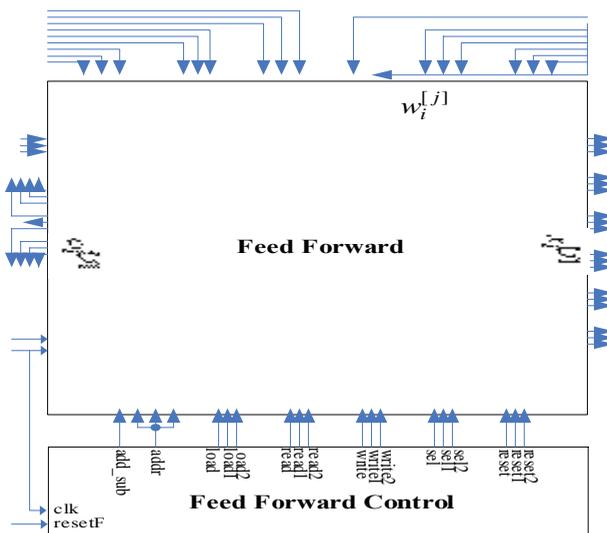
Figure 3. The ANN architecture
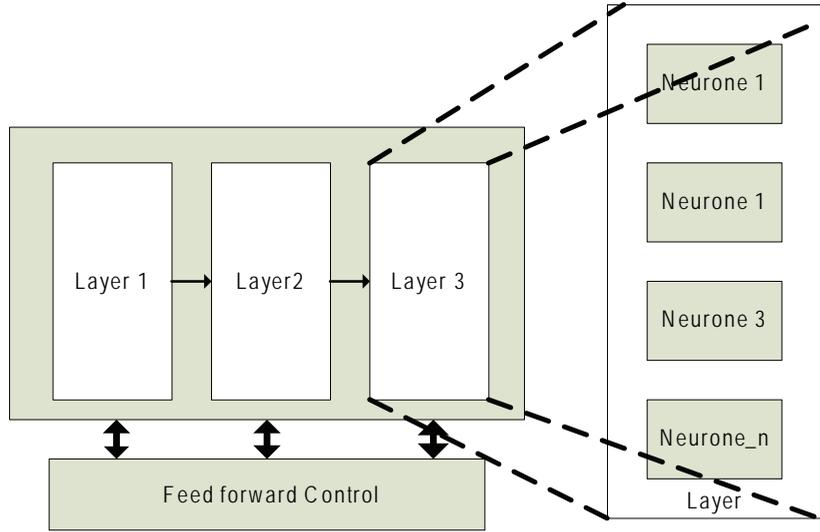
Figure 4. Equivalent hardware architecture of the neuron
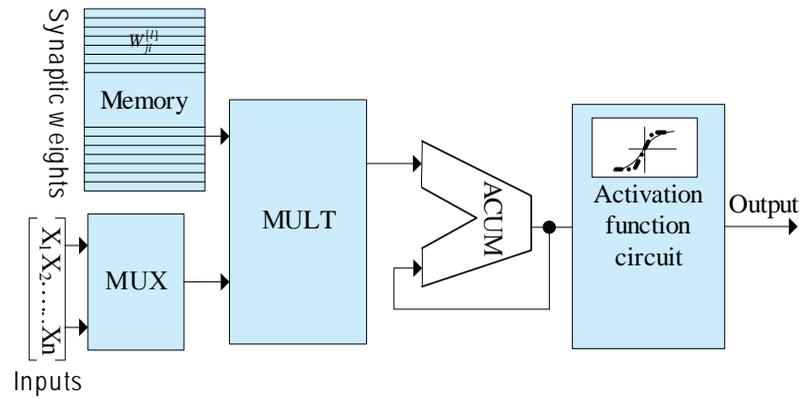
- Each multiplier is connected to a memory. The depth of each memory is equal to the number of neurons constituting the layer
- The whole network is controlled by a control unit module.

Each **circuit** that constructs the neuron is an IP **core** "Intellectual Property" that can be generated from the Xilinx Core Generator.

The feed forward control **module** is composed of three phases: control of the neuron, control of the

layer and control of the network. Considering the fact that neurons work in parallel, so control of the layer is similar to the control of the neuron plus the multiplexer's control. Control of the neuron is divided into four phases: start, initialization, synaptic multiplication/accumulation and storage of the weighted sum. The first state diagram of the feed forward control module which was designed, was based on the Moore machine in which the system vary only when its state change. The drawback of this machine is that it is not generic. For example, (load=0, reset=0) allows the accumulator to add a value present at the input register. This accumulated value must be done as many times as the number of neurons in the previous layer. Thus, if we change the number of neurons from one layer to another one, we have to change all the flow state of the control module. To overcome this problem, the Moor machine is replaced by the Mealy machine in which we add a counter program with a generic value M and a transition variable Max such that:

$$\begin{cases} if output \_ counter \ = M \ \to Max = 1 \\ else \qquad\qquad\qquad \to Max = 0 \end{cases}$$

where the value of M is done equal to the number of neuron.

By using this strategy, we obtain an architecture that has two important key features: generecity and flexibility. Generecity is related to the data word size, precision, and memory depth which are kept as generic parameters at the top level of the VHDL description. The flexibility feature is related to the size of the network (the number of neurons in each layer), thus it is possible to add neurons by simple copy/past of the neurons boxes or cores and it is also possible to remove them by simple cut operation of the boxes. It is also possible to use other IP **cores** from the library (example replace parallel MULT with pipeline MULT) to change the performances of the network without changing the VHDL code. Thus, the **design for reuse** concept is applied.

The Direct On-Chip Implementation Strategy

In this section, we propose the equivalent architecture for implementation of the three successive phases of the back propagation algorithm. Fig.5 depicts the pro-

posed architecture which is composed of a feed forward module, an Error-calculation module and an Update module. The set of the three modules is controlled by a global control unit. The feed forward **module** computes equations (1) and (2). The Error module computes equations (3) and (4) and the Update module computes equations (5) and (6). Each **module** exhibits a high degree of regularity of the structure, modularity and repeatability which make the whole ANN a good candidate for the application of the design for reuse concept. As in the off-chip implementation case, first the unit control unit has been done using a Moore machine that integrates control of the three **modules**: feed forward, error and update modules. In order to achieve reuse, we have replaced the Moore machine by a Mealy machine. Thus, the size of the network can be modified by simple copy/past or remove operations of the boxes.

The Run Time Reconfiguration Strategy

Our strategy for run time reconfiguration follows the following steps: first the feed forward and the global control modules are configured. The results are stored in the Bus macro module of the Virtex FPGA device. In the next step, the feed forward module is reset from the FPGA and the Update and Error modules are configured. The generated results are stored in the Bus macro modules and the same procedure is applied to the next training example of the ANN. A more detailed description is given in (N. Izeboudjen and all, 2007).

Performance Evaluation

In this section, we discuss the performance of the three implementation figures of the back propagation algorithm. The parameters to be considered are the number of configurable logic blocs (CLB), the time response (TR) and the number of Million connexions per second (MCPS). A comparison of these parameters is done between the Virtex-II and Virtex-4 families. Functional simulation is achieved using ModelSim simulator (ModelSim user guide reference). The RTL synthesis is achieved using the Mentor graphics synthesis tool (Mentor Graphics synthesis tool user guide reference) and for final implementation, the ISE foundation place and rout (8.2) tool is used (ISE foundation user guide reference).

Figure 5. Architecture of the BP algorithm
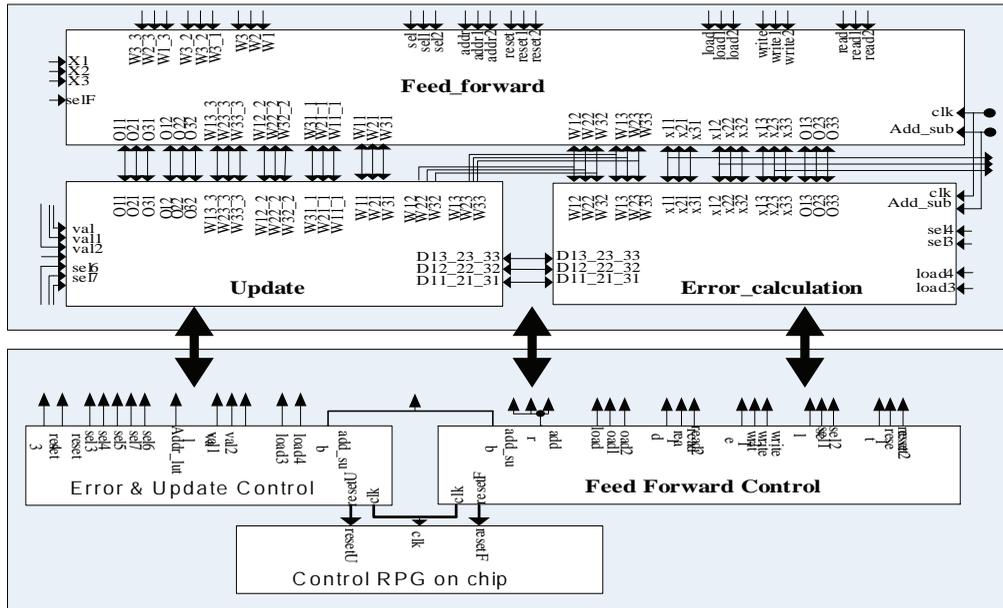
Our first application is an ANN classifier that is used to classify heart coronary diseases. The network has been trained off chip using the MATLAB 6.5 tool. After training the dimension of the network as well as the synaptic weight were fixed. The network has a dimension of (1, 8, 1) and the synaptic weights have a data width of 24 bits. For this application we selected the circuits XC2V1000 and XC4VLX15 devices, of Virtex-II and Virtex-4 respectively. Synthesis results show that the XC2V1000 circuit consume 99% in terms of (CLB), the time response TR = 44.46 (ns) while the MCPS=360. Concerning the XC4VLX15, it consumes 82% in term of CLB, TR= 26.76 (ns) and MCPS= 597. Thus, the XC4VLX15 achieves better performances in term of area (gain 19% of CLB in term of area), the speed rate is 1.6 and MCPS rate is 1.6.

Our second application is the classical (2, 2, 1) "XOR" ANN which is used as a benchmark for non-linearly separable problems. The general on chip learning implementation has been applied to the network. It is to be mentioned that area constraints could not be met for

the first family XC2V1000 as well as the XC4VlX15, and we have tried several families until we fixed the XC4VlX80 for Virtex-4 and the XC2V8000 for Virtex-II. Synthesis results show that the XC2V8000 circuit consume 22% in terms of (CLB), the time response TR= 59.5 (ns) while the MCPS=202. Concerning the XC4VLX80, it consumes 30% in term of (CLB), TR = 47.93 (ns) and MCPS= 250. From these results we can conclude that with the Virtex-II family we can gain 8% of (CLB) in term of area ; this is due to the fact that the Virtex-II integrates more multipliers than the Virtex-4 and in which the MAC component is integrated into the DSP48 (XC4VlX80 has 80 MAC DSP and XC2V8000 has 168 bloc multipliers). But the Virtex-4 **circuit** is faster than the Virtex-II and can achieve more MCPS (rate of ~1.24). The on chip implementation requires a lot of multipliers and this is why, we recommend using it if the timing constraints are not critical.

In the third application, three arbitrary networks are used to show the performance of the (RTR) over the global implementation. These are a (3,3,3) network,

a (16,16,16) network and a (16,64,8) network. The results show that when the size of the network is big it is difficult to implement the whole RPG into one FPGA. With the RTR we can achieve more than 30% reduction in the area and more than 40% increase in speed and MCPS.

FUTURE TRENDS

The proposed ANN environment is still under construction. The design approach is based on the use of pre-designed IP **cores** which are generated from the Xilinx **Core** generator tool. Our next objective is to enrich and enhance the library of the IP cores, especially in the case of implementation of the activation function (sigmoid, linear transfer circuits), and to evaluate and compare the performances of the ANN regarding others pre-designed IP **cores**.

Also, we plan to extend the reuse concept of the ANN to other ANNs algorithms (Kohonen, Hopfield networks)

Concerning the run-time reconfiguration (RTR), the next step is to integrate the RTR design approach with the planeAhead design tool (PlanAhead user guide reference).

As future work, we plan to evaluate and analysis the cost of the design for reuse concept applied to ANNs

CONCLUSION

Through this paper, we have presented a successful design approach for FPGA implementation of ANNs. We have applied a **design for reuse** strategy and parametric design to achieve our goal. The proposed methodology offers high flexibility because the size of the ANN can be changed by simple copy/remove of the neurons **cores**. In addition the format, data widths and precision are considered as generic parameters. Thus, different applications can be targeted in a reduced design time. As for the three applications, the first conclusion is that the new Virtex-4 FPGA devices achieve faster networks comparing to Virtex-II; but regarding to the area; i.e. number of CLBs, the Virtex-II is better. Thus in our opinion, the Virtex-II is well suited as a platform to experiment ANN implementations. This can help to give new directions for future work.

REFERENCES

Amos R. Omondi and Jagath C. rajapakse (2006), "FPGA implementation of neural networks", Springer Verlag.

C.S. Lindsey, Th. Lindblad (1995)" Survey of neural network hardware", SPIE Vol. 2492, pp 1194-1205

C. S. Lindsey and T. Lindblad (1994), "Review of Neural Network Hardware: A user's perspective", IEEE Third Workshop on Neural Networks: from Biology to High Energy Physics.

M. Diepenhorst, M. van Veelen, J.A.G Nijhuis and L. Spaanenburg(1999), IEEE, pp 2302-2305

Freeman, J.A. and Skapura, D. M (1991) "Neural networks Algorithms, Applications and Programming Techniques" Addison Wesley publisher.

ISE Core generator, www.xilinx.com

J. Zhu, G. J. Milne, B. K. Gunther (1999) " Towards an FPGA Reconfigurable Computing Environment for Neural Networks Implementations" Artificial neural networks, Conference publication No 470, IEE , Volume 2, pp 661-666

Keating, Michael; Bricaud, Pierre (2002) "Reuse methodology manual", Kluwer academic publisher.

McCulloch, W.S, & Pitts, W. (1943), "A Logical Calculus of Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics. (5) 115-133.

Model Sim user guide www.model.com

Mentor graphics user guide www.mentor.com

N. Izeboudjen, A.Farah, H. Bessalah, A. Bouridene, N. Chikhi (2007), "Towards a Platform for FPGA Implementation of the MLP Based back Propagation Algorithm" IWANN, LNCS, pp. 497-505

OpenCores: www.opencores.org

PlanAhead User guide www.xilinx.com

Richard P. Lippmann (1984), «An Introduction to computing with neural nets », *IEEE ASSP Magazine*, pp. 4 -22

F. Schumann, S. Hofmann, J. Schemmel, K. Meier, (2002), **"Evolvable Hardware"** Proceedings NASA/DoD Conference on Volume, Issue, pp 266 - 273

Xilinx application notes XAPP290 (2004) "Two Flows for Partial Reconfiguration: Module Based or Difference Based", pp (1-28) www.xilinx.com.

KEY TERMS

ASIC: Acronym Application Specific Integrated Circuits

CLB: Acronym for Configurable Logic Blocs

FPGA: Field Programmable Gate Arrays

High Level Synthesis: A top down design methodology that transform an abstract level such as the VHDL language into a physical implementation level

On-Chip Training: A term that design implementation the three phases of the back propagation algorithm into one or several chips

Off-Chip Training: Training of the network is done using software tools like MATLAB and only the feed forward phase is considered generalisation.

RTL: Acronym of Register Transfer Level

Run Time Reconfiguration: A solution that permits to use the smallest FPGA and to reconfigure it several times during the processing. Run time reconfiguration can be partial or global.

VHDL: Acronym for Very high speed integrated circuits Hardware Description Language)

H

HOPS: A Hybrid Dual Camera Vision System

Stefano Cagnoni
Università degli Studi di Parma, Italy

Monica Mordonini
Università degli Studi di Parma, Italy

Luca Mussi
Università degli Studi di Perugia, Italy

Giovanni Adorni
Università degli Studi di Genova, Italy

INTRODUCTION

Biological vision processes are usually characterized by the following different phases:

- Awareness: natural or artificial agents operating in dynamic environments can benefit from a, possibly rough, global description of the surroundings. In human this is referred to as peripheral vision, since it derives from stimuli coming from the edge of the retina.
- Attention: once an interesting object/event has been detected, higher resolution is required to set focus on it and plan an appropriate reaction. In human this corresponds to the so-called foveal vision, since it originates from the center of the retina (fovea).
- Analysis: extraction of detailed information about objects of interest, their three-dimensional structure and their spatial relationships completes the vision process. Achievement of these goals requires at least two views of the surrounding scene with known geometrical relations. In humans, this function is performed exploiting binocular (stereo) vision.

Computer Vision has often tried to emulate natural systems or, at least, to take inspiration from them. In fact, different levels of resolution are useful also in machine vision. In the last decade a number of studies dealing with multiple cameras at different resolutions have appeared in literature. Furthermore, the ever-growing computer performances and the ever-decreasing cost of video equipment make it possible to develop systems

which rely mostly, or even exclusively, on vision for navigating and reacting to environmental changes in real time. Moreover, using vision as the unique sensory input makes artificial perception closer to human perception, unlike systems relying on other kinds of sensors and allows for the development of more direct biologically-inspired approaches to interaction with the external environment (Trullier 1997).

This article presents HOPS (Hybrid Omnidirectional Pin-hole Sensor), a class of dual camera vision sensors that try to exalt the connection between machine vision and biological vision.

BACKGROUND

In the last decade some investigations on hybrid dual camera systems have been performed (Nayar 1997; Cui 1998; Adorni 2001; Adorni 2002; Adorni 2003; Scotti 2005; Yao 2006). The joint use of a moving standard camera and of a catadioptric sensor provides these sensors with their different and complementary features: while the traditional camera can be used to acquire detailed information about a limited region of interest ("foveal vision"), the omnidirectional sensor provides wide-range, but less detailed, information about the surroundings ("peripheral vision"). Possible employments for this class of vision systems are video surveillance applications as well as mobile robot navigation tasks. Moreover, their particular configuration makes it possible to realize different strategies to control the orientation of the standard camera; for example, scattered focus on different objects permits to perform recognition/classification tasks while continu-

ous movements allow to track any interesting moving object. Three-dimensional reconstruction based on stereo vision is also possible.

HOPS: HYBRID OMNIDIRECTIONAL PIN-HOLE SENSOR

This article is focused on the latest prototype of the HOPS (Hybrid Omnidirectional-Pinhole Sensor) sensor (Adorni 2001; Adorni 2002; Adorni 2003, Cagnoni 2007). HOPS is a dual camera vision system that achieves a high-resolution 360-degrees field of view as well as 3D reconstruction capabilities. The effectiveness of this hybrid sensor derives from the joint use of a traditional camera and a central catadioptric camera which both satisfy the single-viewpoint constraint. Having two different viewpoints from which the world is observed, the sensor can therefore act as a stereo pair finding effective applications in surveillance and robot navigation.

To create a handy versatile system that could meet the requirements of the whole vision process in a wide variety of applications, HOPS has been designed to be considered as a single integrated object: one of the most direct advantages offered by this is that, once it is

assembled and calibrated, it can be placed and moved anywhere (for example in the middle of a room ceiling or on a mobile robot) without any need for further calibrations.

Figure 1 shows the latest two HOPS prototypes. In the one that has been used for the experiments reported here, the traditional camera which, in this version, cannot rotate, has been placed on top and can be pointed downwards with an appropriate fixed tilt angle to obtain a high-resolution view of a restricted region close to the sensor. In the middle, one can see the catadioptric camera consisting of a traditional camera pointing upwards to a hyperbolic mirror hanging over it and held by a plexiglas cylinder. As can be observed, the mirror can be moved up and down to permit optimal positioning (Swaminathan 2001; Strelow 2001) during calibration.

Moreover, to avoid undesired light reflections on the internal surface of the Plexiglas cylinder, a black needle has been placed on the mirror apex as suggested in (Ishiguro 2001). Finally, in the lower part, some circuits generate video synchronization signals and allow for external connections.

The newer version of HOPS (see Figure 1, right) overcomes some limitations of the present one. It uses two digital high-resolution Firewire cameras,

Figure 1. The two latest versions of the HOPS sensor: the one used for experiments (left) and the newest version (right) which is currently being assembled and tested.
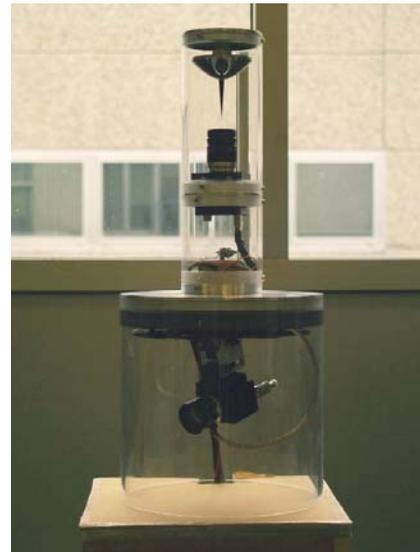
in conjunction with mega-pixel lenses characterized by a very low TV-distortion, to achieve better image quality. Furthermore, in this new version the traditional camera is hung to a stepper motor, controlled via a USB interface, and therefore is able to rotate. This time the traditional camera has been placed below the catadioptric part: this makes it possible to have no wires within the field of view of the omnidirectional image besides allowing, in surveillance applications, to see also the blind area of the omnidirectional view due to the reflection of the camera on the mirror.

Sensor Calibration

In order to extract metric information from two-dimensional images, one must perform a calibration of the camera and estimate the geometric parameters needed to describe image formation. Therefore, after calibration, relationships between points on images and their real position in the 3D space can be expressed by mathematical equations which can solve metric problems.

Sensor calibration can be based on a standard Photogrammetric Calibration (Kraus 1993; Zhang 2000) using a heavily structured environment with grids of points of known coordinates. First, the two cameras are calibrated independently, before assembling them on the sensor, to estimate their intrinsics as well as the radial distortion introduced by the optics. Then, the mirror is accurately positioned with respect to the camera in order to achieve single-viewpoint vision for the catadioptric

part of the sensor as described by (Benosman 2001). The last, but probably most important, phase of the calibration is aimed at detecting geometric relationships between the traditional image and the omnidirectional one: once again, a set of known points was used to estimate the parameters of the mapping.

Notice that the relationships that were computed between the two views are constant in time because of the sensor structure. In this way, once the calibration procedure is over, no external fixed references are needed any longer, and one can place the sensor anywhere and perform stereo vision tasks without needing any further calibration.

Mirror to Camera Positioning

To position the hyperbolic mirror with respect to the standard camera and achieve the single-viewpoint characteristic for the catadioptric part of the sensor, one can operate as follows.

Supposing that the single view-point constraint is satisfied, and since the mirror profile is known, the camera calibration data and some simple equations can be used to calculate the expected projections of any known 3D point set onto the omnidirectional image. To verify the correctness of the relative mirror-to-camera positioning, a calibration box has been built with grids of known coordinates painted on its inner walls. Hence, after placing the sensor into it, the mirror can be manually moved until the grids appearing on the image taken in real time match the theoretical

Figure 2. Mirror position calibration: the sensor inside the calibration box (left) and the acquired omnidirectional image (right) with the correct grid positions superimposed in white.
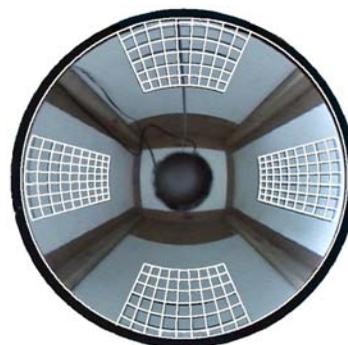
ones super-imposed over it as they should appear if the mirror had been correctly placed (see Figure 2). This is a very cheap method which, however, yields very good results.

Joint Camera Calibration

To obtain a fully effective stereo system it is essential to make a joint camera calibration to extract information about the relative positioning of the camera pair. Usually, the internal reference frame of one of the two cameras is chosen as the global reference for the pair: since two different kinds of cameras are available, the simplest choice is to set the omnidirectional camera's frame as the global reference for the whole sensor. Using once again the above-mentioned grids of points, images pair (omnidirectional and traditional) of grids lying on different (parallel) planes with known relative positions are acquired. Once 3D coordinates of points positions, referred to the sensor reference frame, have been estimated through the omnidirectional image, solving for geometric constraints between points projections in the traditional image permits to estimate the relative position of the traditional camera.

To take the standard camera rotation into consideration, its position has to be described by a more complex transformation than a simply fixed rototranslation: the geometric and kinematic coupling between the two cameras has to be understood and modeled with more parameters. Obviously, this requires that images be taken with the traditional camera in many different positions.

After this joint camera calibration, HOPS can be used to perform metric measurements on the images, obtaining three-dimensional data referred to its own global reference frame: this means that no further calibrations are needed to take sensor displacements into account.

Perspective Reprojections & Inverse Perspective Mapping

One of the opportunities offered by a perspective image is the possibility to apply an Inverse Perspective Mapping (IPM) transformation (Little 1991) to obtain a different image in which the information content is homogeneously distributed among all pixels. Since central catadioptric cameras are characterized by a single viewpoint, the images acquired by them are perspective images suitable to be used for IPM. Choosing a virtual image plane as the new domain for the IPM, a perspective reprojection similar to traditional images can be obtained from part of those omnidirectional images.

Figure 3 shows a pair of images acquired by HOPS and a perspective reconstruction of the omnidirectional view obtained applying an IPM on the corresponding area seen by the traditional camera. As can be noticed, the difference in resolution between the two perspective views is considerable. Choosing a horizontal plane as reference for the IPM, it is possible to obtain something very similar to an orthographic view of that area, usually referred to as "bird's eye view". If the floor is used as reference to perform IPM on both images, it is possible to extract useful information about objects/obstacles surrounding the system (Bertozzi 1998).

3D Reconstruction Tests

To verify the correctness of the calibration process, an estimation of the positions of points in a three-dimensional space can be performed along with other tests. After capturing one image from each of the two views, the points in the test pattern are automatically detected and for each one the light rays from which it was generated are computed based on the projection model obtained during calibration. Since the estimated homologous rays are usually skew lines, the shortest segment joining the two rays can be found and its middle point used as an estimate of the point's 3D position.

In Table 1, results obtained using a 4x3 point test-pattern with 60 mm between point centers are reported. Even if the origin of the sensor reference system is physically inaccessible and no high-precision instruments were available, this pattern was placed as accurately as possible 390 mm perpendicularly ahead of the sensor itself (along the y direction in the chosen reference frame) and centered along the x direction: the z coordinates of the points in the top row were measured to be equal to 55 mm. This set-up is reflected by the estimated values for the first experiment reported in Table 1. More relevantly, the mean distance between points was estimated to be 59.45 mm with a standard deviation $\sigma = 1.14$: those values are fully compatible with the resolution available for measuring distances on the test pattern and with the mirror resolution (also limited by image resolution).

In a second experiment, a test-pattern with six points spaced by 110 mm, located about 1 m ahead, 0.25 m

Figure 3. Omnidirectional image (above, left) and traditional image (above, right) acquired by HOPS. Below a perspective reconstruction of part of the omnidirectional one is shown.
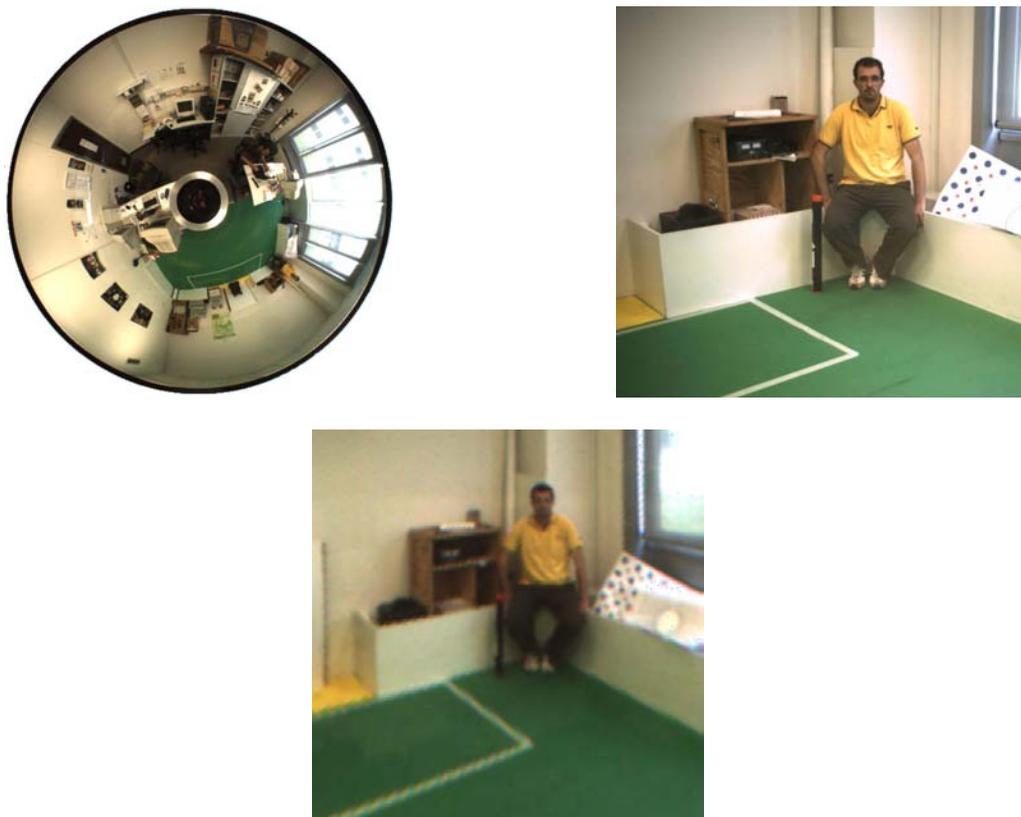
to the right and a bit below the sensor, has been used. In the lower part of Table 1 the estimated positions are shown: the estimated mean distance was 109.09 mm with a standard deviation σ = 8.89. In another test with the same pattern located 1.3 m ahead, 0.6 m to the left and 0.5 m below the sensor (see Figure 4) the estimated mean distance was of about 102 mm with a standard deviation σ = 9.98.

It should be noticed that, at those distances, taking into account image resolution as well as the mirror profile, the sensor resolution is of the same order of magnitude as the errors obtained. Furthermore, the method used to find the center of circles suffers from luminance and contrast variations: substituting circles with adjacent alternate black and white squares and

using a corner detector capable of sub-pixel accuracy would probably yield better results.

FUTURE TRENDS

A field which nowadays draws great interest is autonomous vehicle navigation. Even if at the moment there are still many problems to be solved before seeing autonomous public vehicles, industrial applications are already possible. Since results in omnidirectional visual servoing and ego-motion estimation are also applicable to hybrid dual camera systems, and many more opportunities are offered by the presence of a second high-resolution view, the use of such devices in this field

Table 1. 3D estimation results: the tables show the estimated positions obtained. The diagrams below them show the estimated distances between points on the test-pattern. All values are in mm.

Estimated points positions - Experiment 1			
(-91.21,393.24, 55.17)	(-31.12,394.10, 54.31)	(29.52,390.55, 55.09)	(88.75,389.55, 54.91)
(-92.18,393.89, -5.73)	(-30.94,389.93, -5.05)	(28.91,391.85, -5.27)	(88.68,389.89, -6.35)
(-91.04,393.16,-64.70)	(-31.57,391.47,-64.21)	(27.65,383.38,-61.20)	(87.17,388.22,-63.58)

	60.1		60.75		59.24	
60.91		59.51		60.38		61.26
	61.37		59.88		59.81	
58.99		59.18		56.58		57.27
	59.5		59.85		59.76	

Estimated points positions - Experiment 2		
(122.91,972.18, 15.11)	(223.79,956.70, 16.85)	(324.80,954.04, 16.23)
(125.77,992.39,-94.46)	(231.38,992.42,-95.30)	(330.52,971.28,-90.04)

	102.08		101.05	
111.45		117.95		107.81
	105.61		101.51	

Figure 4. Omnidirectional image (left) and traditional image (right) acquired for a 3D stereo estimation test

is desirable. Even if most applications of these systems are related with surveillance, they could be applied even more directly to robot-aided human activities, since robots/vehicles involved in these situations are less critical and their controllability is easier.

CONCLUSIONS

The Hybrid Omnidirectional Pin-hole Sensor (HOPS) dual camera system has been described. Since its joint camera calibration leads to a fully calibrated hybrid stereo pair from which 3D information can be extracted, HOPS suits several kinds of applications. For example, it can be used for surveillance and robot self-localization or obstacle detection, offering the possibility to integrate stereo sensing with peripheral/foveal active vision strategies: once objects or regions of interest are localized on the wide-range sensor, the traditional camera can be used to enhance the resolution with which these areas can be analyzed.

Tracking of multiple objects/people relying on high-resolution images for recognition and access control or estimating velocity, dimensions and trajectories are some examples of surveillance tasks for which HOPS is suitable. Accurate obstacle detection, landmark localization, robust ego-motion estimation or three-dimensional environment reconstruction are other examples of possible applications related to (autonomous/holonomous) robot navigation in semi-structured or completely unstructured environments. Some preliminary experiments have been performed to solve both surveillance and robot navigation with encouraging results.

REFERENCES

Adorni, G., Bolognini, L., Cagnoni, S., & Mordonini, M. (2001). A non-traditional omnidirectional vision system with stereo capabilities for autonomous robots. In F. Esposito (Ed.), *Lecture Notes In Computer Science* Springer-Verlag, Vol. 2175, 344–355.

Adorni, G., Cagnoni, S., Carletti, M., Mordonini, M. & Sgorbissa, A. (2002). Designing omnidirectional vision sensors. *AI*IA Notizie* 15(1), 27–30.

Adorni, G., Cagnoni, S., Mordonini, M. & Sgorbissa, A. (2003). Omnidirectional stereo systems for robot navigation. *Proceedings of the IEEE Workshop on Omnidirectional Vision*. Madison Wisconsin, 21 June 2003. IEEE Computer Society Press, 78-89.

Benosman, R. & Kang, S. (2001). *Panoramic vision: Sensors, theory and applications.* Springer-Verlag New York, Inc.

Bertozzi, M., Broggi, A. & Fascioli, A. (1998). Stereo inverse perspective mapping: Theory and applications. *Image and Vision Computing Journal* Elsevier Vol. 16, 585-590.

Cagnoni, S., Mordonini, M., Mussi, L. & Adorni, G. (2007). Hybrid stereo sensor with omnidirectional vision capabilities: Overview and calibration procedures. *Proceedings of the 14th International Conference of Image Analysis and Processing*. Modena, 11-13 September 2007. IEEE Computer Society Press, 99-104.

Cui, Y., Samarasekera, S., Huang, Q. & Greiffenhagen, M. (1998). Indoor monitoring via the collaboration between a peripheral sensor and a fovea1 sensor. *VS: Proceedings of the 1998 IEEE Workshop on Visual Surveillance.* Bombay, 2 January 1998. IEEE Computer Society Press, Vol.00, 2-9.

Ishiguro, H. (2001). Development of low-cost compact omnidirectional vision sensors. In R. Benosman & S. Kang (Eds.), *Panoramic vision: Sensors, theory and applications* Springer-Verlag New York, Inc, 23-28.

Kraus, K. (1993). *Photogrammetry: Fundamentals and standard processes* (4th ed., Vol. 1). Dümmler.

Little, J., Bohrer, S., Mallot, H. & Bülthoff, H. (1991). Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics* Springer-Verlag Vol. 64, 177-185.

Nayar, S. & Boult, T. (1997). Omnidirectional vision systems: 1998 PI report. *Proceedings of the 1997 DARPA Image Understanding Workshop*. New Orleans, 11-14 May 1997. Storming Media, 93-99.

Scotti, G., Marcenaro, L., Coelho, C., Selvaggi, F. & Regazzoni, C. (2005). Dual camera intelligent sensor for high definition 360 degrees surveillance. *IEE Proceedings on Vision Image and Signal Processing.* IEE Press. Vol.152, 250-257.

Swaminathan, R., Grossberg, M. D. & Nayar, S. K. (2001). Caustics of catadioptric cameras. *Proceedings of the 8th International Conference on Computer Vision*.

Vancouver, 9-12 July 2001. IEEE Computer Society Press. Vol.2, 2-9.

Trullier, O., Wiener, S., Berthoz, A. & Meyer, J. (1997). Biologically - based artificial navigation systems: Review and prospects. *Progress in Neurobiology*. Elsevier. Vol. 51, 483–544.

Yao, Y., Abidi, B. & Abidi, M. (2006). Fusion of omnidirectional and PTZ cameras for accurate cooperative tracking. In *AVSS: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*. Sydney, 22-24 November 2006. IEEE Computer Society Press, 46-51.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE Computer Society Press. Vol. 22, 1330-1334.

KEY TERMS

Camera Calibration: A procedure used to obtain geometrical information about image formation in a specific camera essential to relate metric distances on the image to distances in the real word. Anyway, some a priori information is needed to reconstruct the third dimension from only one image.

Holonomous Robot: A robot with an unconstrained freedom of movement with no preferential direction.

This means that, from a standing position, it can move as easily in any direction.

Inverse Perspective Mapping (IPM): A procedure which allows for perspective effect to be removed from an image by homogeneously redistributing the information content of the image plane into a new two-dimensional domain.

Lens Distortion: Optical errors in camera lenses, usually due to mechanical misalignment of its parts, can cause straight lines in the observed scene to appear curved in the captured image. The deviation between the theoretical image and the actual one is mostly to be attributed to lens distortion.

Pin-Hole Camera: A camera that uses a tiny hole (the pin-hole) to convey all rays from the observed scene to the image plane. The smaller the pin-hole, the sharper the picture. Pin-hole cameras achieve a potentially infinite depth of field. Because of its geometric simplicity, the "pin-hole model" is used to describe most traditional cameras.

Single Viewpoint Constraint: When all incoming principal light rays of a lens intersect at a single point, an image with a non-distorted metric content is obtained. In this case all information contained in this image is seen from this view-point.

Visual Servoing: An approach to robot control based on visual perception: a vision system extracts information from the surrounding environment to localize the robot and consequently servoing its position.

Hybrid Dual Camera Vision Systems

Stefano Cagnoni
Università degli Studi di Parma, Italy

Monica Mordonini
Università degli Studi di Parma, Italy

Luca Mussi
Università degli Studi di Perugia, Italy

Giovanni Adorni
Università degli Studi di Genova, Italy

INTRODUCTION

Many of the known visual systems in nature are characterized by a wide field of view allowing animals to keep the whole surrounding environment under control. In this sense, dragonflies are one of the best examples: their compound eyes are made up of thousands of separate light-sensing organs arranged to give nearly a 360° field of vision. However, animals with eyes on the sides of their head have high periscopy but low binocularity, that is their views overlap very little. Differently, raptors' eyes have a central part that permits them to see far away details with an impressive resolution and their views overlap by about ninety degrees. Those characteristics allow for a globally wide field of view and for accurate stereoscopic vision at the same time, which in turn allows for determination of distance, leading to the ability to develop a sharp, three-dimensional image of a large portion of their view.

In mobile robotics applications, autonomous robots are required to react to visual stimuli that may come from any direction at any moment of their activity. In surveillance applications, the opportunity to obtain a field of view as wide as possible is also a critical requirement. For these reasons, a growing interest in omnidirectional vision systems (Benosman 2001), which is still a particularly intriguing research field, has emerged. On the other hand, requirements to be able to carry out object/pattern recognition and classification tasks are opposite, high resolution and accuracy and low distortion being possibly the most important ones. Finally, three-dimensional information extraction can be usually achieved by vision systems that combine the use of at least two sensors at the same time.

This article presents the class of hybrid dual camera vision systems. This kind of sensors, inspired by existing visual systems in nature, combines an omnidirectional sensor with a perspective moving camera. In this way it is possible to observe the whole surrounding scene at low resolution, while, at the same time, the perspective camera can be directed to focus on objects of interest with higher resolution.

BACKGROUND

There are essentially two ways to observe a very wide area. It is possible to use many cameras pointed on non-overlapping areas or, conversely, a single camera with a wide field of view. In the former case, the amount of data to be analyzed is much bigger than in the latter one. In addition, calibration and synchronization problems for the camera network have to be faced. On the other hand, in the second approach the system is cheaper, easy to calibrate, while the analysis of a single image is straightforward. In this case, however, the disadvantage is a loss of resolution at which objects details are seen, since a wider field of view is projected onto the same area of the video sensor and thus described with the same amount of pixel as for a normal one. This was clear since the mid 1990s with the earlier experiments with omnidirectional vision systems. Consequently a number of studies on omnidirectional sensors "enriched" with at least one second source of environmental data arose to achieve wide fields of view without loss of resolution. For example some work, oriented to robotics applications, has dealt with a catadioptric camera working in conjunction with a laser scanner as, to cite

only few recent, in (Kobilarov 2006; Mei 2006). More surveillance application-oriented work has involved multi-camera systems, joining omnidirectional and traditional cameras, while other work dealt with geometric aspects of hybrid stereo/multi-view relations, as in (Sturm 2002; Chen 2003).

The natural choice to develop a cheap vision system with both omni-sight and high-detail resolution is to couple an omnidirectional camera with a moving traditional camera. In the sequel, we will focus on this kind of systems that are usually called "hybrid dual camera systems".

Omnidirectional Vision

There are two ways to obtain omnidirectional images. With a special kind of lenses mounted on a standard camera, called "fisheye lenses", it is possible to obtain a field of view up to about 180-degrees in both directions. The widest fisheye lens ever produced featured a 220-degrees field of view. Unfortunately, it is very difficult to design a fisheye lens that satisfies the single viewpoint constraint. Although images acquired by fisheye lenses may prove to be good enough for some visualization applications, the distortion compensation issue has not been solved yet, and the high unit-cost is a major drawback for its wide-spread applications.

Combining a rectilinear lens with a mirror is the other way to obtain omnidirectional views. In the so called "catadioptric lenses" a convex mirror is placed in front of a rectilinear lens achieving a field of view possibly even larger than with a fisheye lens. Using particularly shaped mirrors precisely placed with

respect to the camera is also possible to satisfy the single viewpoint constraint and thus to obtain an image which is perspectively correct. Moreover, catadioptric lenses are usually cheaper than fisheye ones. In Figure 1 a comparison between these two kinds of lenses can be seen.

OVERVIEW OF HYBRID DUAL CAMERA SYSTEMS

The first work concerning hybrid vision sensors is probably the one mentioned in (Nayar 1997) referred to as "Omnidirectional Pan/Tilt/Zoom System" where the PTZ unit was guided by inputs obtained from the omnidirectional view. The next year (Cui 1998) presented a distributed system for indoor monitoring: a peripheral camera was calibrated to estimate the distance between a target and the projection of the camera on the floor. In this way, they were able to precisely direct the foveal sensor, of known position, to the target and track it. A hybrid system for obstacle detection in robot navigation was described in (Adorni 2001) few years later. In this work, a catadioptric camera was calibrated along with a perspective one as a single sensor: its calibration procedure permitted to compute an Inverse Perspective Mapping (IPM) (Little 1991) based on a reference plane, the floor, for both images and hence, thanks to the cameras' disparity, to detect obstacles by computing the difference between the two images. While this was possible only within the common field of view of the two cameras, awareness or even tasks such as ego-motion estimation were potentially pos-

Figure 1. Comparison between image formation in fisheye lenses (left) and catadioptric lenses (right)

Figure 2. A pair of images acquired with the hybrid system described in (Cagnoni 2007). The omnidirectional image (left) and the perspective image (right). The different resolution of the two images is clearly visible.

sible thanks to the omni-view. This system was further improved and mainly tested in RoboCup[1] applications, (Adorni 2002; Adorni 2003; Cagnoni 2007). In Figure 2 it is possible to see a pair of images acquired with such a system.

Some recent work has concentrated on using dual camera systems for surveillance applications. In (Scotti 2005), when some alarm is detected on the omni-directional sensor, the PTZ camera is triggered and the two views start to track the target autonomously. Acquired video sequences and other metadata, like object classification information, are then used to update a distributed database to be queried later by users. Similarly in (Yao 2006), after the PTZ camera is triggered by the omnidirectional one, the target is tracked independently on the two views, but then a modified Kalman filter is used to perform data fusion: this approach achieves an improved tracking accuracy and permits to resolve occasional occlusions leading to a robust surveillance system.

FUTURE TRENDS

Nowadays public order keeping, private property access control and security video surveillance are reasons for which we need to surveil wide areas of our environment. Surveillance is an ever growing market and automatic surveillance is an interesting challenge: many projects are oriented in this direction and in some of them an important role is already played by hybrid dual camera systems. The monitoring system installed between Eagle Pass, Texas, and Piedras Negras, Mexico, by engineers of the Computer Vision and Robotics Laboratory at the University of California, San Diego, affiliated with the California Institute for Telecommunications and Information Technology, is an example of a very complex surveillance system in which hybrid dual camera systems are involved (Hagen 2006). Because of the competitive cost, the compactness and the opportunities offered by these systems, they are likely to be used more and more in the future in intelligent surveillance systems.

Another field subjected to great interest is autonomous vehicle navigation. Even if at the moment there are still many problems to be solved before seeing autonomous public vehicles, industrial applications are already possible. Since omnidirectional visual servoing and ego-motion estimation can actually be implemented also using hybrid dual camera systems, and many more opportunities are offered by the presence of a second high-resolution view, their future involvement in this field is desirable.

CONCLUSIONS

The class of hybrid dual camera systems has been described and briefly overviewed. The joint use of a standard camera and of a catadioptric sensor provides

H

this kind of sensors with their different and complementary features: while the traditional camera can be used to acquire detailed information about a limited region of interest ("foveal vision"), the omnidirectional sensor provides wide-range, but less detailed information about the surroundings ("peripheral vision").

Tracking of multiple objects/people relying on high-resolution images for recognition and access control or estimating object/people velocity, dimensions and trajectory are some examples of possible automatic surveillance tasks for which hybrid dual camera systems are suitable. Furthermore, their use in (autonomous) robot navigation, allows for accurate obstacle detection, egomotion estimation and three-dimensional environment reconstruction. With one of these sensors on board, a mobile robot can be provided with all the necessary information needed to navigate safely in a dynamic environment.

REFERENCES

Adorni, G., Bolognini, L., Cagnoni, S. & Mordonini, M. (2001). A non-traditional omnidirectional vision system with stereo capabilities for autonomous robots. In F. Esposito (Ed.), Springer-Verlag. *Lecture Notes In Computer Science* Vol. 2175, 344–355.

Adorni, G., Cagnoni, S., Carletti, M., Mordonini, M. & Sgorbissa, A. (2002). Designing omnidirectional vision sensors. *AI*IA Notizie* XV (1), 27–30.

Adorni, G., Cagnoni, S., Mordonini, M. & Sgorbissa, A. (2003). Omnidirectional stereo systems for robot navigation. *In Proceedings of the IEEE Workshop on Omnidirectional Vision*. Madison, Wisconsin, 21 June 2003. IEEE Computer Society Press, 79-89.

Benosman, R. & Kang, S. (2001). *Panoramic vision: Sensors, theory and applications.* Springer-Verlag.

Cagnoni, S., Mordonini, M., Mussi, L. & Adorni, G. (2007). Hybrid stereo sensor with omnidirectional vision capabilities: Overview and calibration procedures. In *Proceedings of the 14th International Conference of Image Analysis and Processing* Modena, 11-13 September 2007. IEEE Computer Society Press, 99-104.

Chen, X., Yang, J. & Waibel, A. (2003). Calibration of a hybrid camera network. In *Proceedings of the 9th*

IEEE International Conference on Computer Vision. Nice, 13-16 October 2003. IEEE Computer Society Press, 150-155.

Cui, Y., Samarasekera, S., Huang, Q. & Greiffenhagen, M. (1998). Indoor monitoring via the collaboration between a peripheral sensor and a fovea1 sensor. In *VS: Proceedings of the 1998 IEEE Workshop on Visual Surveillance.* Bombay, 2 January 1998. IEEE Computer Society Press. Vol.00, 2-9.

Hagen, D. & Ramsey, D. (2006). *UCSD engineers deploy novel video surveillance system on Texas Bridge over Rio Grande.* Retrieved June 6, 2007, from the California Institute for Telecommunications and Information Technology Web site: http://www.calit2.net/newsroom/release.php?id=873

Kobilarov, M., Hyams, J., Batavia, P. & Sukhatme, G. S. (2006). People tracking and following with mobile robot using an omnidirectional camera and a laser. In *Proceedings of the IEEE International Conference on Robotics and Automation.* Orlando, 15-19 May 2006. IEEE Computer Society Press, 557-562.

Little, J., Bohrer, S., Mallot, H. & Bülthoff, H. (1991). Inverse perspective mapping simplifies optical flow computation and obstacle detection. In *Biological Cybernetics* Springer-Verlag. Vol.64, 177-185.

Mei, C. & Rives, P. (2006). Calibration between a central catadioptric camera and a laser range finder for robotic applications. In *Proceedings of the IEEE International Conference on Robotics and Automation.* Orlando, 15-19 May 2006. IEEE Computer Society Press, 532-537.

Nayar, S. & Boult, T. (1997). Omnidirectional vision systems: 1998 PI report. In *Proceedings of the 1997 DARPA Image Understanding Workshop.* New Orleans, 11-14 May 1997. Storming Media, 93-99.

Scotti, G., Marcenaro, L., Coelho, C., Selvaggi, F. & Regazzoni, C. (2005). Dual camera intelligent sensor for high definition 360 degrees surveillance. In *IEE Proceedings on Vision Image and Signal Processing*, IEE Press, Vol.152, 250-257.

Sturm, P. (2002). Mixing catadioptric and perspective cameras. In *Proceedings of the Workshop on Omnidirectional Vision.* Copenhagen, 12-14 June 2002. IEEE Computer Society Press, 37-44.

Yao, Y., Abidi, B. & Abidi, M. (2006). Fusion of omnidirectional and PTZ cameras for accurate cooperative tracking. In *AVSS: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance.* Sydney, 22-24 November 2006. IEEE Computer Society Press, 46-51.

KEY TERMS

Camera Calibration: A procedure used to obtain geometrical information about image formation in a specific camera. After calibration, it is possible to relate metric distances on the image to distances in the real word. In any case only one image is not enough to reconstruct the third dimension and some a priori information is needed to accomplish this capability.

Catadioptric Camera: A camera that uses in conjunction catoptric, reflective, lenses (mirrors) and dioptric, refractive, lenses. Usually the purpose of these cameras is to achieve a wider field of view than the one obtained by classical lenses. Even if the field of view of a lens could be improved with any convex surface mirror, those of greater interest are conic, spherical, parabolic and hyperbolic-shaped ones.

Central Catadioptric Camera: A camera that combines lenses and mirrors to capture a wide field of view through a central projection (i.e. a single viewpoint). Most common examples use paraboloidal or hyperboloidal mirrors. In the former case a telecentric lens is needed to focalize parallel rays reflected by the mirror and there are no constraints for mirror to camera relative positioning: the internal focus of the parabola acts as the unique viewpoint; in the latter case it is possible to use a normal lens, but mirror to camera positioning is critical for achieving a single viewpoint: it is essential that the principal point of the lens coincides with the external focus of the hyperboloid to let the internal one be the unique viewpoint for the observed scene.

Omnidirectional Camera: A camera able to see in all directions. There are essentially two different methods to obtain a very wide field of view: the older one involves the use of a special type of lens, usually referred to as fisheye lens, while the other one uses in conjunction rectilinear lenses and mirrors. Lenses obtained in the latter case are usually called catadioptric lenses and the camera-lens ensemble is referred to as catadioptric camera.

PTZ Camera: A camera able to pan left and right, tilt up and down, and zoom. It is usually possible to freely control its orientation and zooming status at a distance through a computer or a dedicated control system.

Stereo Vision: A visual perception process that exploits two different views to achieve depth perception. The difference between the two images, usually referred to as binocular disparity, is interpreted by the brain (or by an artificial intelligent system) as depth.

Single Viewpoint Constraint: To obtain an image with a non-distorted metric content, it is essential that all incoming principal light rays of a lens intersect at a single point. In this case a fixed viewpoint is obtained and all the information contained in an image is seen from this point.

ENDNOTE

[1] Visit http://www.robocup.org for more information.

Hybrid Meta–Heuristics Based System for Dynamic Scheduling

Ana Maria Madureira
Polytechnic Institute of Porto, Portugal

INTRODUCTION

The complexity of current computer systems has led the software engineering, distributed systems and management communities to look for inspiration in diverse fields, e.g. robotics, artificial intelligence or biology, to find new ways of designing and managing systems. Hybridization and combination of different approaches seems to be a promising research field of computational intelligence focusing on the development of the next generation of intelligent systems.

A manufacturing system has a natural dynamic nature observed through several kinds of random occurrences and perturbations on working conditions and requirements over time. For this kind of environment it is important the ability to efficient and effectively adapt, on a continuous basis, existing schedules according to the referred disturbances, keeping performance levels. The application of Meta-Heuristics to the resolution of this class of **dynamic scheduling** problems seems really promising.

In this article, we propose a hybrid Meta-Heuristic based approach for complex scheduling with several manufacturing and assembly operations, in dynamic Extended Job-Shop environments. Some self-adaptation mechanisms are proposed.

BACKGROUND

Scheduling Problem

The planning of **Manufacturing Systems** involves frequently the resolution of a huge amount and variety of combinatorial optimisation problems with an important impact on the performance of manufacturing organisations. Examples of those problems are the sequencing and scheduling problems in manufacturing management, routing and transportation, layout design and timetabling problems.

Scheduling can be defined as the assignment of time-constrained jobs to time-constrained resources within a pre-defined time framework, which represents the complete time horizon of the schedule. An admissible schedule will have to satisfy a set of constraints imposed on jobs and resources. So, a scheduling problem can be seen as a decision making process for operations starting and resources to be used. A variety of characteristics and constraints related with jobs and production system, such as operation processing time, release and due dates, precedence constraints and resource availability, can affect scheduling decisions (Leung, 2004) (Brucker, 2004) (Blazewicz, Ecker &Trystrams, 2005) (Pinedo, 2005).

Real world scheduling requirements are related with complex systems operated in dynamic environments. This means that they are frequently subject to several kinds of random occurrences and perturbations, such as new job arrivals, machine breakdowns, employees sickness, jobs cancellation and due date and time processing changes, causing prepared schedules becoming easily outdated and unsuitable. Scheduling under this environment is known as dynamic.

Dynamic scheduling problems may be classified under deterministic, when release times and all other parameters are known and fixed, and under **non-deterministic** when some or all system and job parameters are uncertain, such as when jobs arrive randomly to the system, over time.

Traditional **heuristic** scheduling methods, encounter great difficulties when they are applied to some real-world situations. This is for three main reasons. Firstly, traditional scheduling methods use simplified and deterministic theoretical models, where all problem data are known before scheduling starts. However, many real world optimization problems are dynamic and non-deterministic and, in which changes may occur continually. In practice, static scheduling is not able to react dynamically and rapidly in the presence of dynamic information not previously foreseen in the current schedule.

Secondly, most of the approximation methods proposed for the Job-Shop Scheduling Problems (JSSP) are oriented methods, i.e. developed specifically for the problem in consideration. Some examples of this class of methods are the priority rules and the Shifting Bottleneck (Pinedo, 2005).

Finally, traditional scheduling methods are essentially centralized in the sense that all the computations are carried out in a central computing and logic unit. All the information concerning every job and every resource has to go through this unit. This centralized approach is especially susceptible to problems of tractability, because the number of interacting entities that must be managed together is large and leads to a combinatorial explosion. Particularly since, a detailed schedule is generated over a long time horizon, and planning and execution are carried out in discrete buckets of time. Centralized scheduling is therefore large, complex, and difficult to maintain and reconfigure. On the other hand, the inherent nature of much industrial and service process is distributed. Consequently, traditional methods are often too inflexible, costly, and slow to satisfy the needs of real-world scheduling systems.

By exploiting problem-specific characteristics, classical optimisation methods are not enough for the efficient resolution of those problems or are developed for specific situations (Leung, 2004) (Brucker, 2004) (Logie, Sabaz & Gruver, 2004) (Blazewicz, Ecker &Trystrams, 2005) (Pinedo, 2005).

Meta-Heuristics

As a major departure from classical techniques, a **Meta-heuristic** (MH) method implies higher-level strategy controlling lower-level **heuristic** methods. Meta-heuristics exploit not only the problem characteristics but also ideas based on **artificial intelligence** rationale, such as different types of memory structures and learning mechanisms, as well as the analogy with other optimization methods found in nature.

The interest of the Meta-Heuristic approaches is that they converge, in general, to satisfactory solutions in an effective and efficient way (computing time and implementation effort). The family of MH includes, but it is not limited to Tabu Search, Simulated Annealing, Soft Computing, Evolutionary Algorithms, Adaptive Memory procedures, Scatter Search, Ant Colony Optimization, Swarm Intelligence, and their hybrids.

For literature on this subject, see for example (Glover & Gary, 2003) and (Gonzalez, 2007).

In last decades, there has been a significant level of research interest in **Meta-Heuristic** approaches for solving large real world scheduling problems, which are often complex, constrained and dynamic. Scheduling algorithms that achieve good or near optimal solutions and can efficiently adapt them to perturbations are, in most cases, preferable to those that achieve optimal ones but that cannot implement such an adaptation. This is the case with most algorithms for solving the so-called static scheduling problem for different setting of both single and multi-machine systems arrangements. This reality, motivated us to concentrate on tools, which could deal with such dynamic, disturbed scheduling problems, even though, due to the complexity of these problems, optimal solutions may not be possible to find.

Several attempts have been made to modify algorithms, to tune them for optimization in a changing environment. It was observed in manufacturing all these studies, that the dynamic environment requires an algorithm to maintain sufficient diversity for a continuous adaptation to the changes of the landscape. Although the interest in optimization algorithms for dynamic optimization problems is growing and a number of authors have proposed an even greater number of new approaches, the field lacks a general understanding as to suitable benchmark problems, fair comparisons and measurement of algorithm quality (Branke, 1999) (Cowling & Johanson, 2002) (Madureira, 2003), Madureira, Ramos & Silva, 2004) (Aytug, Lawley, McKay, Mohan & Uzsoy, 2005).

In spite of all the previous trials scheduling problem still known to be NP-complete. This fact incites researchers to explore new directions.

Hybrid Intelligent Systems

Hybridization of intelligent systems is a promising research field of computational intelligence focusing on combinations of multiple approaches to develop the next generation of intelligent systems. An important stimulus to the investigations on Hybrid Intelligent Systems area is the awareness that combined approaches will be necessary if the remaining tough problems in **artificial intelligence** are to be solved. Meta-Heuristics, Bio-Inspired Techniques, Neural computing, Machine Learning, Fuzzy Logic Systems, Evolution-

ary Algorithms, Agent-based Methods, among others, have been established and shown their strength and drawbacks. Recently, hybrid intelligent systems are getting popular due to their capabilities in handling several real world complexities involving imprecision, uncertainty and vagueness (Boeres, Lima, Vinod & Rebello, 2003), (Madureira, Ramos & Silva, 2004) (Bartz-Beielstein, Blesa, Blum, Naujoks, Roli, Rudolph &Sampels, 2007).

HYBRID META-HEURISTICS BASED SCHEDULING SYSTEM

The purpose of this article is to describe an framework based on combination of **Meta-Heuristics**, Tabu Search(TS) and Genetic Algorithms(GA), and constructive optimization methods for solving a class of real world scheduling problems, where the products (jobs) to be processed have due dates, release times and different assembly levels. This means that parts to be assembled may be manufactured in parallel, i.e. simultaneously.

The problem, focused in this work, which we call Extended Job-Shop Scheduling Problem (EJSSP) has major extensions and differences in relation to the classic Job-Shop Scheduling Problem. In this work, we define a job as a manufacturing order for a final item, that could

be Simple or Complex. It may be Simple, like a part, requiring a set of operations to be processed. Complex Final Items, requiring processing of several operations on a number of parts followed by assembly operations at several stages, are also dealt with. Moreover, in practice, scheduling environment tends to be dynamic, i.e. new jobs arrive at unpredictable intervals, machines breakdown, jobs can be cancelled and due dates and processing times can change frequently (Madureira, 2003) (Madureira, Ramos & Silva, 2004).

It starts focusing on the solution of the dynamic deterministic EJSSP problems. For solving these we developed a framework, leading to a dynamic scheduling system having as a fundamental scheduling tool, a hybrid scheduling system, with two main pieces of intelligence (Figure 1).

One such piece is a combination of TS and GA based method and a mechanism for inter-machine activity coordination. The objective of this mechanism is to coordinate the operation of machines, taking into account the technological constraints of jobs, i.e. job operations precedence relationships, towards obtaining good schedules. The other piece is a **dynamic adaptation** module that includes mechanisms for neighbourhood/population regeneration under dynamic environments, increasing or decreasing it according new job arrivals or cancellations.

Figure 1. Hybrid meta-heuristics based scheduling system
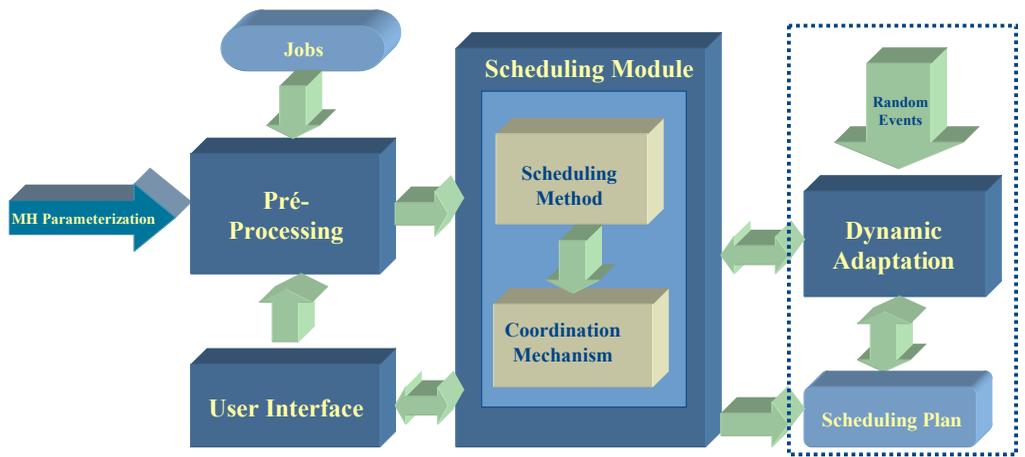
A detailed description of the approach, methods and of its application to concrete problems can be found in Madureira (2003).

Pre-Processing Module

The pre-processing module deals with processing input information, namely problem definition and instantiation of algorithm components and parameters, such as, the initial solution and neighbourhood generation mechanisms, size of neighbourhood/population, tabu list attributes and tabu list length.

Hybrid Scheduling Module

Initially, we start by decomposing the deterministic EJSSP problem into a series of deterministic Single Machine Scheduling Problems (SMSP). We assume the existence of different and known job release times r_j, prior to which no processing of the job can be done and, also, job due dates d_j. Based on these, release dates and due dates are determined for each SMSP and, subsequently, each such problem is solved independently by a TS or a GA(considering a self-parameterization issue). Afterwards, the solutions obtained for each SMSP are integrated to obtain a solution to the main EJSSP problem instance.

The integration of the SMSP solutions may give an unfeasible schedule to the EJSSP. This is why schedule repairing may be necessary to obtain a feasible solution. The repairing mechanism named Inter-Machine Activity Coordination Mechanism (IMACM) carries this out. The repairing is based on coordination of machines activity, having into account job operation precedence and other problem constraints. This is done keeping job allocation order, in each machine, unchanged. The IMACM establishes the starting and the completion times for each operation. It ensures that the starting time for each operation is the higher of the two following values:

- the completion time of the immediately precedent operation in the job, if there is only one, or the highest of all if there are more;
- the completion time of the immediately precedent operation on the machine.

Dynamic Adaptation Module

For non-deterministic problems some or all parameters are uncertain, i.e. are not fixed as we assumed in the deterministic problem. Non-determinism of variables has to be taken into account in real world problems. For generating acceptable solutions in such circumstances our approach starts by generating a predictive schedule, using the available information and then, if perturbations occur in the system during execution, the schedule may have to be modified or revised accordingly, i.e. rescheduling/**dynamic adaptation** is performed. Therefore, in this process, an important decision must be taken, namely that of deciding if and when rescheduling should happen. The decision strategies for rescheduling may be grouped into three categories: continuous, periodic and hybrid rescheduling. In the continuous one rescheduling is done whenever an event modifying the state of the system occurs. In periodic rescheduling, the current schedule is modified at regular time intervals, taking into account the schedule perturbations that have occurred. Finally, for the hybrid rescheduling the current schedule is modified at regular time intervals if some perturbation occurs.

In the scheduling system for EJSSP, dynamic adaptation is necessary due to two classes of events:

- Partial events which imply variability in jobs or operations attributes such as processing times, due dates and release times.
- Total events which imply variability in neighbourhood structure, resulting from either new job arrivals or job cancellations.

While, on one hand, partial events only require redefining job attributes and re-evaluation of the objective function of solutions, total events, on the other hand, require a change on solution structure and size, carried out by inserting or deleting operations, and also re-evaluation of the objective function. Therefore, under a total event, the modification of the current solution is imperative. In this work, this is carried out by mechanisms described in (Madureira, Ramos & Silva, 2004) for SMSP.

Considering the processing times involved and the high frequency of perturbations, rescheduling all jobs from the beginning should be avoided. However, if

work has not yet started and time is available, then an obvious and simple approach to rescheduling would be to restart the scheduling from scratch with a new modified solution on which takes into account the perturbation, for example a new job arrival. When there is not enough time to reschedule from scratch or job processing has already started, a strategy must be used which adapts the current schedule having in consideration the kind of perturbation occurred.

The occurrence of a partial event requires redefinition of job attributes and a re-evaluation of the schedule objective function. A change in job due date requires the re-calculation of the operation starting and completion due times of all respective operations. However, changes in the operation processing times only requires re-calculation of the operation starting and completion due times of the succeeding operations. A new job arrival requires definition of the correspondent operation starting and completion times and a regenerating mechanism to integrate all operations on the respective single machine problems. In the presence of a job cancellation, the application of a regenerating mechanism eliminates the job operations from the SMSP where they appear. After the insertion or deletion of positions, neighbourhood regeneration is done by updating the size of the neighbourhood and ensuring a structure identical to the existing one. Then the scheduling module can apply the search process for better solutions with the new modified solution.

Job Arrival Integration Mechanism

When a new job arrives to be processed, an integration mechanism is needed. This analyses the job precedence graph that represents the ordered allocation of machines to each job operation, and integrates each operation into the respective single machine problem. Two alternative procedures could be used for each operation: either randomly select one position to insert the new operation into the current solution/chromosome or use some intelligent mechanism to insert this operation in the schedules, based on job priority, for example.

Job Elimination Mechanism

When a job is cancelled, an eliminating mechanism must be implemented so the correspondent position/gene will be deleted from the solutions.

Regeneration Mechanisms

After integration/elimination of operations is carried out, by inserting/deleting positions/genes in the current solution/chromosome, population regeneration is done by updating its size. The population size for SMSP is proportional to the number of operations.

After dynamic adaptation process, the scheduling method could be applied and search for better solutions with the modified solution.

In this way we proposed a hybrid system in which some self-organization aspects could be considered in accordance with the problem being solved: the method and/or parameters can change in run-time, the used MH can change according with problem characteristics, etc.

FUTURE TRENDS

Considering the complexity inherent to the manufacturing systems, the dynamic scheduling is considered an excellent candidate for the application of agent-based technology. A natural evolution to the approach above proposed is a Multi-agent Scheduling System that assumes the existence of several Machines Agents (which are decision-making entities) distributed inside the Manufacturing System that interact and cooperate with other agents in order to obtain optimal or near-optimal global performances.

The main idea is that from local, autonomous and often conflicting behaviours of the agents a global solution emerges from a community of machine agents solving locally their schedules and cooperating with other machine agents (Madureira, Gomes & Santos, 2006). Agents must be able to learn and manage their internal behaviours and their relationships with other agents, by **cooperative** negotiation in accordance with business policies defined by user manager. Some self-organization aspects could be considered in accordance with the problem being solved: the method and/or parameters can change in run-time, the agents can use different MH according with problem characteristics, etc.

CONCLUSION

This article proposes a system architecture that makes good use and combination of the advantages of two different Meta-Heuristics: Tabu Search and Genetic Algorithms.

We believe that a new contribution for the resolution of more realistic scheduling problems, the Extended Job-Shop Problems was described. The particularity of our approach is the procedure to schedule operations, as each machine will first find local optimal or near optimal solutions, succeeded by the interaction with other machines trough cooperation mechanisms as a way to find an optimal global schedule, on dynamic environments.

The proposed system is prepared to use other Local Search Meta-Heuristics, to drive schedules based on practically any performance measure and it is not restricted to a specific type of scheduling problems.

REFERENCES

Aytug, Haldun, Lawley, Mark A., McKay, Kenneth, Mohan, Shantha & Uzsoy, Reha(2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, Volume 16 (1), 86-110.

Bartz-Beielstein, Thomas, Blesa, M.J., Blum, C., Naujoks, B., Roli, A., Rudolph, G. & Sampels, M.(2007). *Hybrid Metaheuristics*. Proceedings of 4th International Workshop H. Dortmund, Germany, Lecture Notes in Computer Science. Vol. 4771, ISBN: 978-3-540-75513-5.

Blazewicz. Jacek, Ecker, Klaus H.&Trystram, Denis(2005), Recent advances in scheduling in computer and manufacturing systems. *European Journal of Operational Research*, 164(3), 573-574.

Boeres, Cristina, Lima, Alexandre, Vinod, E.&Rebello, F.(2003). Hybrid Task Scheduling: Integrating Static and Dynamic Heuristics. *15th Symposium on Computer Architecture and High Performance Computing,* 199.

Branke, J.(1999). Evolutionary Approaches to Dynamic Optimization Problems – A Survey. *GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, 34-137.

Brucker, Peter(2004). *Scheduling Algorithms*. Springer, 4rd edition.

Cowling, P.&Johansson, M.(2002). Real time information for effective dynamic scheduling. *European Journal of Operational Research*,139 (2), 230-244.

Glover, Fred & Gary, A. Kochenberger(2003). *Handbook of Metaheuristics*. International Series in Operations Research & Management Science, Springer, Vol. 57, ISBN: 978-1-4020-7263-5.

Gonzalez, Teofilo F.(2007). *Handbook of Approximation Algorithms and Metaheuristics*. Chapman&Hall/Crc Computer and Information Science Series.

Leung, Joseph.(2004). *Handbook of Scheduling*. Chapman&Hall/CRC, Boca Raton, FL.

Logie, S., Sabaz, D. & Gruver, W.A.(2004). Sliding Window Distributed Combinatorial Scheduling using JADE. *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, Netherlands, 1984-1989.

Madureira, Ana(2003). *Meta-Heuristics Application to Scheduling in Dynamic Environments of Discrete Manufacturing*. PhD Dissertation, University of Minho, Braga, Portugal(in portuguese).

Madureira, Ana, Gomes, Nuno & Santos, Joaquim(2006). Cooperative Negotiation Mechanism for Agent Based Distributed Manufacturing Scheduling. *WSEAS Transactions on Systems*, Issue 12, Volume 5, ISSN:1109-2777, 2899-2904.

Madureira, Ana, Ramos, Carlos & Silva, Sílvio(2004). Toward Dynamic Scheduling Through Evolutionary Computing. *WSEAS Transactions on Systems*, Issue 4, Volume 3, 1596-1604.

Pinedo, M.(2005). *Planning and Scheduling in Manufacturing and Services*, Springer-Verlag, New York, ISBN:0-387-22198-0.

KEY TERMS

Cooperation: The practice of individuals or entities working together with common goals, instead of working separately in competition, and in which the success of one is dependent and contingent upon the success of the other.

H

Dynamic Scheduling Systems: Are frequently subject to several kinds of random occurrences and perturbations, such as new job arrivals, machine breakdowns, employee's sickness, jobs cancellation and due date and time processing changes, causing prepared schedules becoming easily outdated and unsuitable.

Evolutionary Computation: A subfield of artificial intelligence that involve techniques implementing mechanisms inspired by biological evolution such as reproduction, mutation, recombination, natural selection and survival of the fittest.

Genetic Algorithms: Particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover.

Hybrid Intelligent Systems: Denotes a software system which employs, a combination of Artificial Intelligence models, methods and techniques, such Evolutionary Computation, Meta-Heuristics, Multi-Agent Systems, Expert Systems and others.

Meta-Heuristics: Form a class of powerful and practical solution techniques for tackling complex, large-scale combinatorial problems producing efficiently high-quality solutions.

Multi-Agent Systems: A system composed of several agents, collectively capable of solve complex problems in a distributed fashion without the need for each agent to know about the whole problem being solved.

Scheduling: Can be seen as a decision making process for operations starting and resources to be used. A variety of characteristics and constraints related with jobs and machine environments (Single Machine, Parallel machines, Flow-Shop and Job-Shop) can affect scheduling decisions.

Tabu Search: A approximation method, belonging to the class of local search techniques, that enhances the performance of a local search method by using memory structures (Tabu List).

A Hybrid System for Automatic Infant Cry Recognition I

Carlos Alberto Reyes-García
Instituto Nacional de Astrofísica Óptica y Electrónica, Mexico

Ramon Zatarain
Instituto Tecnologico de Culiacan, Mexico

Lucia Barron
Instituto Tecnologico de Culiacan, Mexico

Orion Fausto Reyes-Galaviz
Universidad Autónoma de Tlaxcala, Mexico

INTRODUCTION

Crying in babies is a primary communication function, governed directly by the brain; any alteration on the normal functioning of the babies' body is reflected in the cry (Wasz-Höckert, *et al*, 1968). Based on the information contained in the cry's wave, the infant's physical state can be determined; and even pathologies in very early stages of life detected (Wasz-Höckert, *et al,* 1970).

To perform this detection, a Fuzzy Relational Neural Network (FRNN) is applied. The input features are represented by fuzzy membership functions and the links between nodes, instead of weights, are represented by fuzzy relations (Reyes, 1994). This paper, as the first of a two parts document, describes the Infant Cry Recognition System´s architecture as well as the FRNN model. Implementation and testing are reported in the complementary paper.

BACKGROUND

The pioneer works on infant cry were initiated by Wasz-Hockert since the beginnings of the 60's. In one of those works his research group showed that the four basic types of cry can be identified by listening: pain, hunger, pleasure and birth. Further studies led to the development of conceptual models that describe the anatomical and physiologic basis of the production and neurological control of crying (Bosma, Truby & Antolop, 1965). Later on, Wasz-Hockert (1970) applied spectral analysis to identify several types of crying. Other works showed that there exist significant differences among the several types of crying, like healthy infant's cry, pain cry and pathological infant's cry. In one study, Petroni used Neural Networks (Petroni, Malowany, Johnston, and Stevens, 1995) to differentiate between pain and no-pain crying. Cano directed several works devoted to the extraction and automatic classification of acoustic characteristics of infant cry. In one of those studies, in 1999 Cano presented a work where he demonstrates the utility of the Kohonen's Self-Organizing Maps in the classification of Infant Cry Units (Cano-Ortiz, Escobedo-Becerro, 1999) (Cano, Escobedo and Coello, 1999). More recently, in (Orozco, & Reyes, 2003) we reported the classification of cry samples from deaf and normal babies with feed-forward neural networks. In 2004 Cano and his group, in (Cano, Escobedo, Ekkel, 2004) reported a radial basis network (RBN) to find out relevant aspects concerned with the presence of Central Nervous System (CNS) diseases. In (Suaste, Reyes, Diaz, and Reyes, 2004) we showed the implementation of a Fuzzy Relational Neural Network (FRNN) for Detecting Pathologies by Infant Cry Recognition.

The study of connectionist models also known as Artificial Neural Networks (ANN) has enjoyed a resurgence of interest after its demise in the 60's. Research was focused on evaluating new neural networks for pattern classification, training algorithms using real speech data, and on determining whether parallel neural network architectures can be designed to perform efficiently the work required by complex

speech recognition algorithms (Lippmann, 1990). In the connectionist approach, pattern classification is done with a multi-layer neural network. A weight is assigned to every link between neurons in contiguous layers. In the input layer each neuron receives one of the features present in the input pattern vectors. Each neuron in the output layer corresponds to each speech unit class (word or sub-word). The neural network associates input patterns to output classes by modeling the relationship between the two pattern sets. The pattern is estimated or learned by the network with a representative sample of input and output patterns (Morgan, and Scofield, 1991) (Pedrycz, 1991).. In order to stabilize the perceptron's behavior, many researchers had been trying to incorporate fuzzy set theory into neural networks. The theory of fuzzy sets, developed by Zadeh in 1965 (Zadeh, 1965), has since been used to generalize existing techniques and to develop new algorithms in pattern recognition. Pal (Pal, 1992a) suggested that to enable systems to handle real-life situations, fuzzy sets should be incorporated into neural networks, and, that the increase in the amount of computation required with its incorporation, is offset by the potential for parallel computation with high flexibility that fuzzy neural networks have. Pal proposes how to do data fuzzification, the general

system architecture of a fuzzy neural network and the use of 3n-dimensional vectors to represent the fuzzy membership values of the input features to the primary linguistic properties *low, medium,* and *high* (Pal, 1992a) and (Pal, and Mandal, 1992b). On the other side, the idea of using a relational neural network as a pattern classifier was developed by Pedrycz and presented in (Pedrycz, 1991). As a result of the combination of the Pal's and Pedrycz's proposed methodologies in 1994 C. A. Reyes (1994) developed the hybrid model known as fuzzy relational neural network (FRNN).

THE AUTOMATIC INFANT CRY RECOGNITION PROCESS

The infant cry automatic classification process is, in general, a pattern recognition problem, similar to Automatic Speech Recognition (ASR) (Huang, Acero, Hon, 2001). The goal is to take the wave from the infant's cry as the input pattern, and at the end obtain the kind of cry or pathology detected on the baby (Cano, Escobedo and Coello, 1999) (Ekkel, 2002). Generally, the process of Automatic Infant Cry Recognition is done in two steps. The first step is known as signal processing, or feature extraction, whereas the second is known as

Figure 1. Automatic infant cry recognition process

861
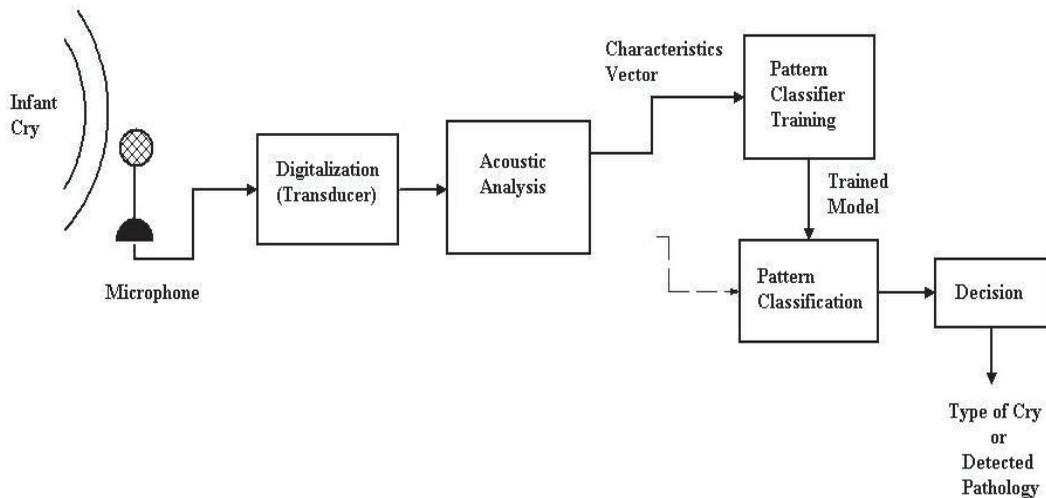
pattern classification. In the acoustical analysis phase, the cry signal is first normalized and cleaned, and then it is analyzed to extract the most important features in function of time. The set of obtained features is represented by a vector, which represents a pattern. The set of all vectors is then used to train the classifier. Later on, a set of unknown feature vectors is compared with the acquired knowledge to measure the classification output efficiency. Figure 1 shows the different stages of the described recognition process.

Cry Patterns Classification

The vectors, representing patterns, obtained in the extraction stage are later used in the classification process. There are four basic schools for the solution of the pattern classification problem, those are: *a)* Pattern comparison (dynamic programming), *b)* Statistic Models (Hidden Markov Models HMM), *c)* Knowledge based systems (expert systems), and *d)* Connectionists Models (neural networks). In recent years, a new strong trend of more robust hybrid classifiers has been emerging. Some of the better known hybrid models result from the combination of neural and fuzzy approaches (Jang, 1993) (Lin Chin-Teng, and George Lee, 1996). For the work shown here, we have implemented a hybrid model of this type, called the Fuzzy Relational Neural Network, whose parameters are found trough the application of genetic algorithms. We selected this kind of model, because of its adaptation, learning and knowledge representation capabilities. Besides, one of its main functions is to perform pattern recognition.

In an Automatic Infant Cry Classification System, the goal is to identify a model of an unknown pattern obtained after the original sound wave is acoustically analyzed, and its dimensionality reduced. So, in this phase we determine the class or category to which each cry pattern belongs to. The collection of samples, each of which is represented by a vector of *n* features, is divided in two subsets: The training set and the test set. First, the training set is used to teach the classifier to distinguish between the different crying types. Then the test set is used to determine how well the classifier assigns the corresponding class to a pattern by means of the classification scheme generated during training.

THE FUZZY NEURAL NETWORK MODEL

The system proposed in this work is based upon fuzzy set operations in both; the neural network's structure and the learning process. Following Pal's idea of a general recognizer (Pal, S.K., 1992a), the model is divided in two main parts, one for learning and another for processing, as shown in Figure 2.

Fuzzy Learning

The fuzzy learning section is composed by three modules, namely the Linguistic Feature Extractor (LFE), the Desired Output Estimator (DOE), and the Neural Network Trainer (NNT). The Linguistic Feature Extractor takes training samples in the form of *n*-dimensional vectors containing *n* features, and converts them to *Nn*-dimensional form vectors, where *N* is the number of linguistic properties. In this case the linguistic properties are *low, medium,* and *high.* The resulting *3n*-dimensional vector is called Linguistic Properties Vector (LPV). In this way an input pattern $\mathbf{F_i} = [F_{i1}, F_{i2}, ...,F_{in}]$ containing *n* features, may be represented as (Pal, and Mandal, 1992b)

$$F_i = \left[\mu_{low(F_{i1})}(F_i), \mu_{med(F_{i1})}(F_i), \mu_{high(F_{i1})}(F_i),...,\mu_{high(F_{in})}(F_i) \right]$$

The DOE takes each vector from the training samples and calculates its membership to class *k*, in an *l*-class problem domain. The vector containing the class membership values is called the Desired Vector (DV). Both LPV and DV vectors are used by the neural Network Trainer (NNT), which takes them for training the network.

The neural network has only one input and one output layer. The input layer is formed by a set of *Nn* neurons, with each of them corresponding to one of the linguistic properties assigned to the *n* input features. In the output layer there are *l* neurons, with each node corresponding to one of the *l* classes; in this implementation, each class represents one type of crying. There is a link from every node in the input layer to every node in the output layer. All the con-

Figure 2. General architecture of the automatic infant cry recognition system

LEARNING

PROCESSING PHASE
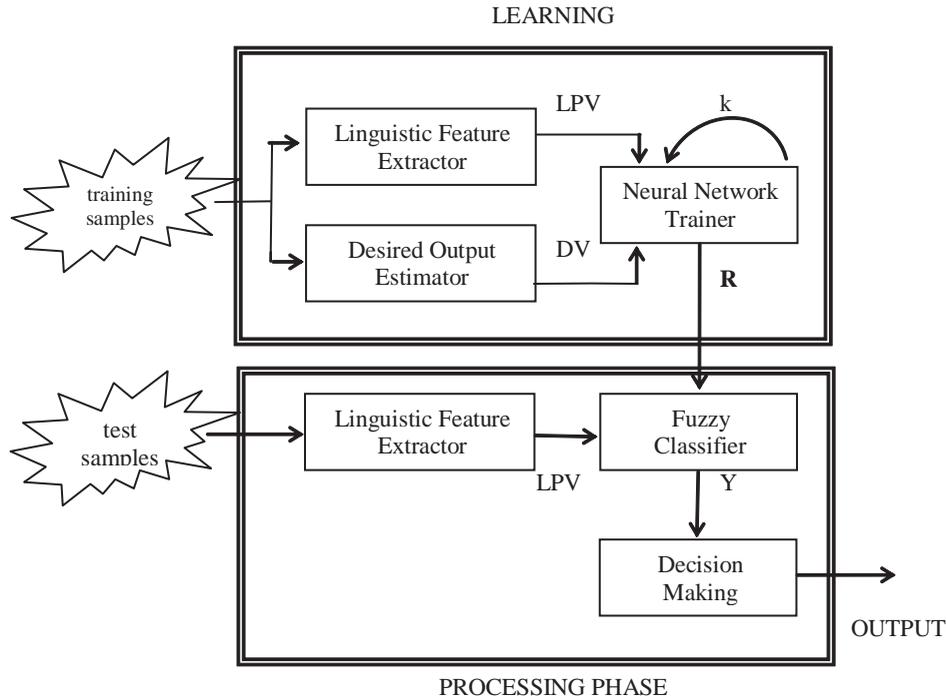
nections are described by means of fuzzy relations *R: X × Y→ [0, 1]* between the input and output nodes. The error is represented by the distance between the actual output and the target or desired output. During each learning step, once the error has been computed, the trainer adjusts the relationship values or weights of the corresponding connections, either until a minimum error is obtained or a given number of iterations are completed. The output of the NNT, after the learning process, is a fuzzy relational matrix (*R* in Figure 1) containing the knowledge needed to further map the unknown input vectors to their corresponding class during the classification process.

Fuzzy Processing

The fuzzy processing section is formed by three different modules, namely the Linguistic Feature Extractor (LFE), the Fuzzy Classifier (FC), and the Decision Making Module (DMM). The LFE works as the one in

the learning phase, described in the previous section. The output of this module is an LPV vector, which along with the fuzzy relational matrix *R*, are used by the Fuzzy Classifier, which obtains the actual outputs from the neural network. The classifier applies the max-min composition to calculate the output. The output of this module is an output vector containing the membership values of the input vector to each of the classes. Finally, the Decision Making module selects the highest value from the classifier and assigns the corresponding class to the testing vector.

Membership Functions

A membership function maps values in a domain to their membership value in a fuzzy set. Several kinds of membership functions are available. In the reported experiments triangular membership functions were used. According to (Park, Cae, and Kandel, 1992) the use of more linguistic properties to describe a pattern

point makes a model more accurate, but too many can make the description unpractical. So, here we use seven linguistic properties: *very low, low, more or less low, medium, more or less high, high,* and *very high*.

Desired Membership Values

Before defining the output membership function, we define the equation to calculate the weighted distance of the training pattern F_j to the kth class in an l-class problem domain as in (Pal, 1992a)

$$z_{ik} = \sqrt{\sum_{j=1}^{n} \left[\frac{F_{ij} - \sigma_{kj}}{\upsilon_{kj}} \right]^2} ,: for\ k = 1, \ldots, l$$

where Fij is the jth feature of the ith pattern vector, σ_{kj} denotes the mean, and υ_{kj} denotes the standard deviation of the jth feature for the kth class. The membership value of the ith pattern to class k is defined as follows

$$\mu_k (F_i) = \frac{1}{1 + \left(\frac{z_{ik}}{f_d} \right) f_e} ,: \mu_k (F_i) \in [0,1]$$

where f_e is the exponential fuzzy generator, and f_d is the denominational fuzzy generator controlling the amount of fuzzines in this class-membership set. In this case, the higher the distance of the pattern from a class, the lower its membership to that class. Since the training data have fuzzy class boundaries, a pattern point usually belongs to more than one class at different degrees.

The Neural Network Trainer

The neural network model discussed here is based on the relational neural structure proposed by Pedrycz in (Pedrycz, W., 1991).

The Relational Neural Network (RNN): Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be a finite set of input nodes and let $\mathbf{Y} = \{y_1, y_2, \ldots, y_l\}$ represent the output nodes set in an l-class problem domain. When the *max-min* composition operator denoted $X \circ R$ is applied to a fuzzy set X and a fuzzy relation R, the output is a new fuzzy set Y, we have

$$Y = X \circ R$$
$$Y(y_j) = \max_{x_i} \left(\min \left(X(x_i), R(x_i, y_j) \right) \right) \tag{1}$$

where X is a fuzzy set, Y is the resulting fuzzy set and R a fuzzy relation $R : X \times Y \rightarrow [0,1]$ describing all relationships between input and output nodes. We will take the whole neural network represented by expression (1) as a collection of l separate n-input single-output cells.

Learning in a Fuzzy Neural Network: If the actual response from the network does not match the target pattern; the network is corrected by modifying the link weights to reduce the difference between the observed and target patterns. To measure the difference a performance index called equality index is defined, which is

$$T(y) \equiv Y(y) = \begin{cases} 1 + T(y) - Y(y) \, if & Y(y) > T(y) \\ 1 + Y(y) - T(y) \, if & Y(y) < T(y) \\ 1, if & Y(y) = T(y) \end{cases}$$

where $T(y)$ is the target output at node y, and $Y(y)$ is the actual output at the same node. In a problem with n input patterns, there are n input-output pairs (x_{ij}, t_i) where t_i is the target value when the input is X_{ij}.

Parameters Updating: Pedricz also proposes to complete the process of learning separately for each output node. The learning algorithm is a version of the back-propagation algorithm. Let's consider an n-input-L-output neural network having the following form

$$y_i = f(x_i; a, \upsilon) = \left(\bigvee_{j=1}^{n} (a_j \wedge x_{ij}) \right)$$

where $\mathbf{a} = [a_1, a_2, \ldots, a_L]$ is a vector containing all the weights or relations, $\mathbf{x_i} = [x_{i1}, x_{i2}, \ldots, x_{in}]$ is the vector with the values observed in the input nodes. The parameters a and υ are updated iteratively by taking increment Δa_m resulting from deviations between all pairs y_i and t_i as follows

$$a(k+1) = a(k) + \Psi_1(k) \left[\frac{\Delta a(k+1)}{Nn} + \eta \frac{\Delta a(k)}{Nn} \right]$$

where k is the learning step. Ψ_1 and Ψ_2 are non-increasing functions of k controlling the decreasing influence of increments Δa_m. Ψ is the learning momentum

specifying the level of modification of the learning parameters with regard to their values in the previous learning step k. A way of determining the increments Δa_m is with regard to the mth coordinates of a, $m = 1, 2,..., L$. The computation of the overall performance index, and the derivatives to calculate the increments for each coordinate of a, and v are explained in detail in (Reyes, C. A., 1994). Once the training has been terminated, the output of the trainer is the updated relational matrix, which will contain the knowledge needed to map unknown patterns to their corresponding classes.

FUTURE TRENDS

One unexplored possibility of improving the FRNN performance is the use of other fuzzy relational products instead of max-min composition. Moreover, membership functions have parameters which can be optimized by genetic algorithms any other optimizing technique. Adequate parameters may improve learning and recognition efficiency of the FRNN.

CONCLUSIONS

We have presented the development and implementation of an AICR system as well as a powerful hybrid classifier, the FRNN, which is a model formed by the combination of fuzzy relations and artificial neural networks. The synergistic symbiosis obtained though the fusion of both methodologies will be demonstrated. In the related paper on applications of this model, we will show some practical results, as well as an improved model by means of genetic algorithms.

ACKNOWLEDGMENTS

REFERENCES

Bosma, J. F., Truby, H. M., and Antolop, W. (1965), *Cry Motions of the Newborn Infant*. Acta Paediatrica Scandinavica (Suppl.), 163, 61-92.

Cano, Sergio D, Escobedo, Daniel I., and Coello, Eddy (1999), El Uso de los Mapas Auto-Organizados de Kohonen en la Clasificación de Unidades de Llanto Infantil, Grupo de Procesamiento de Voz, 1er Taller AIRENE, Universidad Catolica del Norte, Chile, pp 24-29.

Cano, Sergio D, Escobedo, Daniel I., Ekkel, Taco (2004) A Radial Basis Function Network Oriented for Infant Cry Classification, Proc. of 9th Iberoamerican Congress on Pattern Recognition, Puebla, Mexico.

Cano-Ortiz, S.D,. Escobedo-Becerro, D. I (1999), Clasificación de Unidades de Llanto Infantil Mediante el Mapa Auto-Organizado de Kohoeen, I Taller AIRENE sobre Reconocimiento de Patrones con Redes Neuronales, Universidad Católica del Norte, Chile,pp. 24-29.

Ekkel, T. (2002), *Neural Network-Based Classification of Cries from Infants Suffering from Hypoxia-Related CNS Damage*, Master Thesis, University of Twente. The Netherlands.

Huang, X., Acero, A., Hon, H. (2001) *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice-Hall, Inc., USA.

Jang, J.-S. R. (1993), ANFIS: Adaptive Network-based Fuzzy Inference System, in IEEE Transactions on Systems, Man, and Cybernetics, 23 (03):665-685.

Lin Chin-Teng, and George Lee, C.S. (1996), Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems, Prentice Hall, Upper Saddle River, NJ.

Lippmann, R.P. (1990), Review of Neural Networks for Speech Recognition", in Readings in Speech Recognition, Morgan Kauffman Publishers Inc., San Mateo, Calif, pp 374-392.

Morgan, D.P., and Scofield, C.L. (1991), *Neural Networks and Speech Processing*, Kluwer Academic Publishers, Boston.

Orozco, J., Reyes, C.A. (2003), Mel-frequency Cepstrum Coefficients Extraction from Infant Cry for Classification of Normal and Pathological Cry whit Feed-Forward Neural Networks, Proc. of ESANN, Bruges, Belgium.

Pal, S.K. (1992a) Multilayer Perceptron, Fuzzy Sets, and Classification", in IEEE Trans. on Neural Networks, vol 3, No 5, Sep 1992, pp 683-697.

Pal, S.K. and Mandal, D.P. (1992b), Linguistic Recognition Systems Based on Approximated Reasoning, in Information Science, vol. 61, No 2, pp 135-161.

Park, D., Cae, Z., and Kandel, A. (1992), Investigations on the Applicability of Fuzzy Inference, in Fuzzy Sets and Systems, vol 49, pp 151-169.

Pedrycz, W.(1991), Neuro Computations in Relational Systems, *IEEE Trans .On Pattern Analysis and Intelligence*, vol. 13, No 3, pp 289-296.

Petroni, M., Malowany, A. S., Johnston, C., and Stevens, B. J., (1995),. Identification of pain from infant cry vocalizations using artificial neural networks (ANNs), The International Society for Optical Engineering. Volume 2492. Part two of two. Paper #: 2492-79. pp.729-738.

Reyes, C. A., (1994) *On the design of a fuzzy relational neural network for automatic speech recognition*, Doctoral Dissertation, The Florida State University, Tallahassee, Fl,. USA.

Suaste, I., Reyes, O.F., Diaz, A., Reyes, C.A. (2004) Implementation of a Linguistic Fuzzy Relational Neural Network for Detecting Pathologies by Infant Cry Recognition, Proc. of IBERAMIA, Puebla, Mexico , pp. 953-962.

Wasz-Höckert, O., Lind, J., Vuorenkoski, V., Partanen, T., & Valanne, E. (1970) *El Llanto en el Lactante y su Significación Diagnóstica*, Cientifico-Medica, Barcelona.

Wasz-Höckert, O., Lind, J., Vuorenkoski, V., Partenen, T., Valanne, E. (1968), The infant cry: a spectrographic and auditory analisis, Clin. Dev. Med. 29, pp. 1-42

Zadeh, L.A. (1965), Fuzzy Sets, Inform. Contr., vol 8, pp 338-353.

KEY TERMS

Artificial Neural Networks: A network of many simple processors that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Automatic Infant Cry Recognition (AICR): A process where the crying signal is automatically analyzed, to extract acoustical features looking to determine the infant's physical state, the cause of crying or even detect pathologies in very early stages of life.

Back propagation Algorithm: Learning algorithm of ANNs, based on minimising the error obtained from the comparison between the outputs that the network gives after the application of a set of network inputs and the outputs it should give (the desired outputs).

Fuzzy Relational Neural Network (FRNN): A hybrid classification model combining the advantages of fuzzy relations with artificial neural networks.

Fuzzy Sets: A generalization of ordinary sets by allowing a degree of membership for their elements. This theory was proposed by Lofti Zadeh in 1965. Fuzzy sets are the base of fuzzy logic.

Hybrid Intelligent System: A software system which employs, in parallel, a combination of methods and techniques from Soft Computing.

Learning Stage: A process to teach classifiers to distinguish between different pattern types.

A Hybrid System for Automatic Infant Cry Recognition II

Carlos Alberto Reyes-García
Instituto Nacional de Astrofísica Óptica y Electrónica, Mexico

Sandra E. Barajas
Instituto Nacional de Astrofísica Óptica y Electrónica, Mexico

Esteban Tlelo-Cuautle
Instituto Nacional de Astrofísica Óptica y Electrónica, Mexico

Orion Fausto Reyes-Galaviz
Universidad Autónoma de Tlaxcala, Mexico

INTRODUCTION

Automatic Infant Cry Recognition (AICR) process is basically a problem of pattern processing, very similar to the Automatic Speech Recognition (ASR) process (Huang, Acero, Hon, 2001). In AICR first we perform acoustical analysis, where the crying signal is analyzed to extract the more important acoustical features, like; LPC, MFCC, etc. (Cano, Escobedo and Coello, 1999). The obtained characteristics are represented by feature vectors, and each vector represents a pattern. These patterns are then classified in their corresponding pathology (Ekkel, 2002). In the reported case we are automatically classifying cries from normal, deaf and asphyxiating infants.

We use a genetic algorithm to find several optimal parameters needed by the Fuzzy Relational Neural Network FRNN (Reyes, 1994), like; the number of linguistic properties, the type of membership function, the method to calculate the output and the learning rate. The whole model has been tested on several data sets for infant cry classification. The process, as well as some results, is described.

BACKGROUND

In the first part of this document a complete description of the AICR system as well as of the FRNN is given. So, with continuity purposes, in this part we will concentrate in the description of the genetic algorithm and the whole system implementation and testing.

A genetic algorithm refers to a model introduced and investigated by John Holland (John Holland, 1975) and by students of Holland (DeJong, 1975). Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad. Recently, numerous papers and applications combining fuzzy concepts and genetic algorithms (GAs) have become known, and there is an increasing concern in the integration of these two topics. In particular, there are a great number of publications exploring the use of GAs for developing or improving fuzzy systems, called genetic fuzzy systems (GFSs) (Cordon, Oscar, *et al*, 2001) (Casillas, Cordon, del Jesus, Herrera, 2000).

EVOLUTIONARY DESIGN

Within the evolutionary techniques, perhaps one of the most popular is the genetic algorithm (AG) (Goldberg, 1989). Its structure presents analogies with the biological theory of evolution, and is based on the principle of the survival of the fittest individual (Holland, 1975). Generally, a genetic algorithm has five basic components (Michalewicz, 1992). A representation of potential solutions to the problem, a form to create potential initial solutions, a fitness function that is in charge to evaluate solutions, genetic operators that alter the offspring's composition, and values for parameters like the size of the population, crossover probability, mutation probability, number of generations and others. Here we present different features of the genetic

algorithm used to find a combination of parameters for the FRNN.

Chromosomal Representation

The binary codification is used in genetic algorithms, and Holland in (Holland, 1975) gave a theoretical justification to use it. Holland argued that the binary codification allows having more schemes than a decimal representation. Scheme is a template that describes a subgroup of strings that share certain similarities in some positions throughout their length (Goldberg, 1989). The problem variables consist of the number of linguistic properties, the type of membership function, the classification method and the learning rate. We are interested in having between 3 and 7 linguistic properties, so, the number of linguistic variables is encoded into a binary string of 3 bit length. The membership function is represented as a 2 bit string, where [00] decodes the Trapezoidal membership function, [01] decodes the Π function, [10] decodes the Triangular function, [11] decodes the Gaussian membership function. The classification methods are also coded as a 2 bit string, where [00] represents the max-min composition, [01] represents the geometrical mean and [10] represents the relational square product. Finally, the learning rate is represented as a binary string of 3 bit length, where [000] decodes to 0.1 learning rate, [001] decodes to 0.2 learning rate, [010] decodes to 0.31 learning rate, [011] decodes to 0.4 learning rate, and [100] decodes to 0.5 learning rate. A larger learning rate is not desirable, so all other bit values are ignored. The chromosome is obtained by concatenating all the above strings. Figure 1 shows an example of the chromosomal representation. Initial population is generated from a random selection of chromosomes, a population size of 50 was considered.

Genetic Operations

We use four genetic operations, namely elitism, roulette wheel selection, crossover and mutation. ***Elitism***: In order to ensure that the members with highest fitness value of the population stay in the next generation we apply elitism. It has been demonstrated (Günter, Rudolph, 1994), that a genetic algorithm must use elitism to be able to show convergence. At each iteration of the genetic algorithm we select the members with the four highest fitness values and we put them in the next generation.

Selection: In the genetic algorithm the selection process is made in a probabilistic way, it is to say, the less apt individuals even have a certain opportunity to be selected. There are many different types of selection approaches; we use the roulette wheel selection, where members of the population have a probability of being selected that is directly proportionate to their fitness. ***Crossover***: In this work we use a single point crossover. Observing the performance of different crossover operators, De Jong (De Jong, K., 1975) concluded that, although increasing the number of points of crosses affects its schemes from a theoretical perspective, in practice this does not seem to have a significant impact. The crossover is the principal operator in the genetic algorithm. Based on some experiments we decided to determine the crossover point randomly and the crossover probability was fixed at 0.8. ***Mutation***: This operator allows the introduction of new chromosomal material in the population. We selected a gene randomly and we replaced it by its complement, a zero is changed by a one and a one is changed by a zero. Some authors suggest that the mutation probability equal to $1/L$, where L is the length of the chain of bits is an inferior limit acceptable for the optimal percentage of mutation (Bäck, Thomas, 1993). In this work the mutation probability is fixed at 0.05.

Figure 1. Chromosomal representation

1	0	1	0	0	0	0	0	0	1

| linguistic properties | | membership | | classification | | learning rate |
| function | | method |
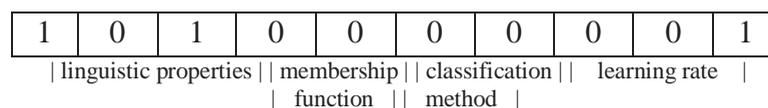
Fitness Function

The objective function of our optimization problem is called fitness function. This function must be able to penalize the solutions that are not good and award the good ones so they can propagate quickly (Coello, Carlos A., 1995). As a fitness function we use the classification error given by the Fuzzy Relational Neural Network. Then the fitness function is defined by the following equation

$$F = e_{FRNN}$$

In this case we define the classification error as follows

$$e_{FRNN} = \frac{No.PM}{No.S}$$

where *No.PM* represents the number of perfect matches, in other words, it represents the number of samples classified correctly. The term *No.S* represents the total number of given samples to the FC.

IMPLEMENTATION AND RESULTS

Signal Processing

The analysis of the raw cry waveform provides the information needed for its recognition. At the same time, it discards unwanted information such as background noise, and channel distortion (Levinson S.E., and Roe, D.B., 1990). Acoustic feature extraction is a transformation of measured data into pattern data. Some of the most important techniques used for analyzing cry wave signals are: Discrete Fourier Transform (DFT), cepstral processing, and Linear Prediction Analysis (LPA) (Ainsworth, W.A., 1988) (Schafer and Rabiner 1990). The application of these techniques during signal processing obtains the values of a set of acoustic features. The features may be spectral coefficients, linear prediction coefficients (LPC), Mel frequency cepstral coefficients (MFCC), among others (Ainsworth, W.A., 1988). The set of values for *n* features may be repre-

sented by a vector in an *n*-dimensional space. Each vector represents a pattern.

For the present experiments we work with samples of infant cries. The infant cries were collected by recordings done directly by medical doctors and then, each signal wave was divided in segments of 1 second, each segment represents a sample. Then, acoustic features were obtained by means of techniques as Linear Prediction Coefficients (LPC) and Mel Frequency Cepstral Coefficients (MFCC), by the use of the freeware program Praat v4.0.8 (Boersma, P., Weenink, 2002). Every sample of 1 second is divided in frames of 50-milliseconds and from each frame we extract 16 coefficients, this procedure generates vectors whit 304 coefficients by sample. In this paper we show the results obtained with Mel Frequency Cepstral Coefficients.

In order to reduce the dimensions of the sample vectors we apply Principal Component Analysis. The FRNN and the genetic algorithm are implemented in Matlab. We have a corpus of 157 samples of normal infant cry, 340 of asphyxia infant cry, and 879 of hypo acoustics. Also we have a corpus of 192 samples of pain and 350 samples of hunger crying. We worked with a population of 50 individuals and the number of training epochs for the FRNN was set at three. The initial population was randomly chosen. The number of generations needed for the genetic algorithm was of only three. These values were set on the basis of the observation of the results of several experiments.

Preliminary Results

Three different classification experiments were made, the first one consists in classifying deaf and normal infant cry, the second one was made to classify infant cry in categories called asphyxia and normal, and the third one to classify hunger and pain crying. In each task the training samples and the test samples are randomly selected. The results of the model in the classification of deaf and normal cry are given in Table I. In Table II we show the results obtained in the second classification task. Finally Table III shows the results in the classification of hunger and pain cry. In every classification task the GA was run about 15 times and the reported results show the average of the best classification in each experiment.

H

Table 1. Results of classifying deaf and normal cry

Characteristics	Successful codification	Interpretation	Accuracy
Number of linguistic properties	0 1 1	3	
Membership function	0 1	Π	98%
Classification method	0 0	max-min	
Learning rate	0 0 1	0.2	

Table 2. Results of classifying asphyxia and normal cry

Characteristics	Successful codification	Interpretation	Accuracy
Number of linguistic properties	0 1 1	3	
Membership function	0 1	Π	84%
Classification method	0 1	geometrical mean	
Learning rate	0 10	0.31	

Table 3. Results of classifying hunger and pain cry

Characteristics	Successful codification	Interpretation	Accuracy
Number of linguistic properties	1 1 1	7	
Membership function	0 1	Π	95.24%
Classification method	0 0	max-min	
Learning rate	0 1 0	0.31	

Performance Comparison with Other Models

Reyes and Orozco (Orozco, Reyes, 2003) classified cry samples from deaf and normal babies, obtaining recognition results around 97.43%. Reyes *et al* (Suaste, Reyes, Diaz, Reyes, 2004) showed an implementation of a linguistic fuzzy relational neural network to classify normal and pathological infant cry with percentage of correct classification of 97.3% and 98%. Petroni, Malowany, Johnston and Stevens (1995) classified cry from normal babies to identify pain with artificial neural networks and report results of correct classification that go from 61% with cascade-correlation networks up to 86.2% with feed-forward neural networks. In (Lederman, 2002) Dror Lederman presents some classification results for infants with respiratory distress syndrome RDS (related to asphyxia) versus healthy infants. For the classification he used a Hidden Markov Model architecture with 8 states and 5 Gaussians/state. The results reported are of 63 % of total mean correct classification.

FUTURE TRENDS

AICR systems may expand their utility by training them to recognize a larger number of pathologies. The first requirement to achieve this goal is to collect a suitable set of labeled samples for any target pathology. The GA presented here optimizes some parameters of the FRNN, but the model has more. So, other parameters can be added to the chromosomal representation in order to improve the model, like initial values of the relational matrix and of the bias vectors, number of training epochs, and the values of the exponential fuzzy generator and the denominational fuzzy generator used by the DOE.

CONCLUSIONS

The proposed genetic algorithm computes a selection of the number of linguistic properties, the membership function used to calculate the linguistic features, the method to calculate the output of the classifier in the fuzzy processing section and the learning rate of the FRNN. The solution obtained by the proposed genetic algorithm is a set of characteristics that the FRNN can use to make the classification of infant cry. The use of linguistic properties allows us to deal with the impreciseness of infant cry and provides the classifier with very useful information. By applying the linguistic information and given the nature of the model, it is not necessary to get training through a high number of learning epochs, a high number of iterations in the genetic algorithm is not necessary either. The results of classifying deaf and normal infant cry are very similar to other models, but when we classify hunger and pain the results are much better than other models.

ACKNOWLEDGMENTS

REFERENCES

Bäck, Thomas (1993), Optimal mutation rates in genetic search, *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, California: Morgan Kaufmann, pp 2-8.

Boersma, P., Weenink (2002), *D. Praat v 4.0.8. A system for doing phonetics by computer*, Institute of Phonetic Sciences of the University of Amsterdam, February.

Bosma, J. F., Truby, H. M., & Antolop, W. (1965), *Cry Motions of the Newborn Infant*. Acta Paediatrica Scandinavica (Suppl.), 163, 61-92.

Cano, Sergio D, Escobedo, Daniel I., and Coello, Eddy (1999), El Uso de los Mapas Auto-Organizados de Kohonen en la Clasificación de Unidades de Llanto Infantil, Grupo de Procesamiento de Voz, 1er Taller AIRENE, Universidad Catolica del Norte, Chile, pp 24-29.

Casillas, J., Cordon, O., Jesus, M.J. del, Herrera, F., (2000), *Genetic Feature Selection in a FuzzyRule. Based Classification System Learning Process for High Dimensional Problems,* Technical Report ·DECSAI-000122, Universidad de Granada, Spain.

Coello, Carlos A.(1995), Introducción a los algoritmos genéticos, Soluciones Avanzadas. Tecnologías de Información y Estrategias de Negocios, Año 3, No. 17, pp. 5-11.

Cordon, Oscar, Herrera, Francisco, Hoffmann, Frank and Magdalena, Luis, (2001), *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases,* Singapore, World Scientific.

De Jong, K. (1975), An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. Dissertation. Dept. of Computer and Communication Sciences, Univ. of Michigan, Ann Arbor.

Ekkel, T. (2002), *Neural Network-Based Classification of Cries from Infants Suffering from Hypoxia-Related CNS Damage*, Master Thesis. University of Twente. The Netherlands.

Goldberg, David E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*. Massachusetts: Addison-Wesley.

Günter, Rudolph (1994), Convergence analysis of canonical genetic algorithms, *IEEE Transactions on Neural Networks*, vol. 5, pp 96-101.

Holland, J. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press.

Huang, X., Acero, A., Hon, H. (2001) *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice-Hall, Inc., USA.

Lederman, D. (2002), "*Automatic Classification of Infants' Cry*". Master Thesis. University of Negev. Israel.

Petroni, M., Malowany, A. S., Johnston, C., and Stevens, B. J., (1995),. Identification of pain from infant cry vocalizations using artificial neural networks (ANNs), The International Society for Optical Engineering. Volume 2492. Part two of two. Paper #: 2492-79. pp.729-738.

Reyes, C. A., (1994), *On the design of a fuzzy relational neural network for automatic speech recognition*, Doctoral Dissertation, The Florida State University, Tallahassee, Fl,. USA.

Suaste, I., Reyes, O.F., Diaz, A., Reyes, C.A. (2004) Implementation of a Linguistic Fuzzy Relational Neural Network for Detecting Pathologies by Infant Cry Recognition, Proc. of IBERAMIA, Puebla, Mexico , pp. 953-962.

Zbigniew Michalewicz (1992), *Genetic algorithms + data structures = evolution programs,* Springer-Verlag, 2nd ed.

KEY TERMS

Binary Chromosome: Is an encoding scheme representing one potential solution to a problem, during a searching process, by means of a string of bits.

Evolutionary Computation: A subfield of computational intelligence that involves combinatorial optimization problems. It uses iterative progress, such as growth or development in a population, which is then selected in a guided random search to achieve the desired end. Such processes are often inspired by biological mechanisms of evolution.

Fitness Function: It is a function defined over the genetic representation and measures the *quality* of the represented solution. The fitness function is always problem dependent.

Genetic Algorithms: A family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information. Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad.

Hybrid Intelligent System: A software system which employs, in parallel, a combination of methods and techniques mainly from subfields of Soft Computing.

Signal Processing: The analysis, interpretation and manipulation of signals. Processing of such signals includes storage and reconstruction, separation of information from noise, compression, and feature extraction.

Soft Computing: A partnership of techniques which in combination are tolerant of imprecision, uncertainty, partial truth, and approximation, and whose role model is the human mind. Its principal constituents are Fuzzy Logic (FL), Neural Computing (NC), Evolutionary Computation (EC) Machine Learning (ML) and Probabilistic Reasoning (PR).

IA Algorithm Acceleration Using GPUs

Antonio Seoane
University of A Coruña, Spain

Alberto Jaspe
University of A Coruña, Spain

INTRODUCTION

Graphics Processing Units (GPUs) have been evolving very fast, turning into high performance programmable processors. Though GPUs have been designed to compute graphics algorithms, their power and flexibility makes them a very attractive platform for general-purpose computing. In the last years they have been used to accelerate calculations in physics, computer vision, artificial intelligence, database operations, etc. (Owens, 2007).

In this paper an approach to general purpose computing with GPUs is made, followed by a description of artificial intelligence algorithms based on Artificial Neural Networks (ANN) and Evolutionary Computation (EC) accelerated using GPU.

BACKGROUND

General-Purpose Computation using Graphics Processing Units (GPGPU) consists in the use of the GPU as an alternative platform for parallel computing taking advantage of the powerful performance provided by the graphics processor (*General-Purpose Computation Using Graphics Hardware Website*; Owens, 2007).

There are several reasons that justify the use of the GPU to do general-purpose computing (Luebke, 2006):

- Last generation GPUs are very fast in comparison with current processors. For instance, a NVIDIA 8800 GTX card has computing capability of approximately 330 GFLOPS, whereas an Intel Core2 Duo 3.0 GHz processor has only a capability of about 48 GFLOPS.
- GPUs are highly-programmable. In the last years graphical chip programming capacities have grown very much, replacing fixed-programming engines with programmable ones, like pixel and vertex engines. Moreover, this has derived in the appearance of high-level languages that help its programming.
- GPUs evolution is faster than CPU's one. The increase in GPU's performance is nowadays from 1.7x to 2.3x per year, whereas in CPUs is about 1.4x. The pressure exerted by videogame market is one of the main reasons of this evolution, what forces companies to evolve graphics hardware continuously.
- GPUs use high-precision data types. Although in the very beginning graphics hardware was designed to work with low-precision data types, at the present time internal calculations are computed using 32 bits float point numbers.
- Graphics cards have low cost in relation to the capacities that they provide. Nowadays, GPUs are affordable for any user.
- GPUs are highly-parallel and they can have multiple processors that allow making high-performance parallel arithmetic calculations.

Nevertheless, there are some obstacles. First, not all the algorithms fit for the GPU's programming model, because GPUs are designed to compute high-intensive parallel algorithms (Harris, 2005). Second, there are difficulties in using GPUs, due mainly to:

- GPU's programming model is different from CPU's one.
- GPUs are designed to graphics algorithms, therefore, to graphics programming. The implementation of general-purpose algorithms on GPU is quite different to traditional implementations.
- Some limitations or restrictions exist in programming capacities. Most functions on GPU's programming languages are very specific and dedicated to make calculations in graphics algorithms.

- GPU's architectures are quite variable due to their fast evolution and the incorporation of new features.

Therefore it is not easy to port an algorithm developed for CPUs to run in a GPU.

Overview of the Graphics Pipeline

Nowadays GPUs make their computations following a common structure called *Graphics Pipeline*. The Graphics Pipeline (Akenine-Möller, 2002) is composed by a set of stages that are executed sequentially inside the GPU, allowing the computing of graphics algorithms. Recent hardware is made up of four main elements. First, the *vertex processors*, that receive vertex arrays from CPU and make the necessary transformations from their positions in space to the final position in the screen. Second, the *primitive assembly* build graphics primitives (for instance, triangles) using information about connectivity between different vertex. Third, in the *rasterizer*, those graphical primitives are discretized and turned into fragments. A fragment represents a potential pixel and contains the necessary information (color, depth, etc.) to generate the final color of a pixel. Finally, in the *fragment processors*, fragments become pixels to which final color is written in a target buffer, that can be the screen buffer or a texture.

In the present, GPUs have multiple vertex and fragment processors that compute operations in parallel. Both are programmable using little pieces of code called vertex and fragment programs, respectively. In the last years different high-level programming languages have released like Cg/HLSL (Mark, 2003; HLSL Shaders) or GLSL (*OpenGL Shading Language Information Site*), that make easier the programming of those processors.

The GPU Programming Model

There is a big difference between programming CPUs and GPUs due mainly to their different programming models. GPUs are based on the *stream programming model* (Owens, 2005a; Luebke, 2006; Owens, 2007), where all data are represented by a *stream* that can be defined as a sorted set of data of the same type. A *kernel* operates on full streams, and takes input data from one or more streams to produce one or more output streams. The main characteristic of a kernel is

that it operates on the whole stream, instead individual elements. The typical use of a kernel is the evaluation of a function over each element from an input stream, calling this a *map* operation. Other operations of a kernel are expansions, reductions, filters, etc. (Buck, 2004; Horn, 2005; Owens, 2007). The kernel generated outputs are always based on their input streams, what means that inside the kernel, the calculations made on an element never depends of the other ones. In stream programming model, applications are built connecting multiple kernels. An application can be represented as a dependency graph where each graph node is a kernel and each edge represents a data stream between kernels (Owens, 2005b; Lefohn, 2005).

The behavior of graphic pipeline is similar to the stream programming model. Data flows through each stage, where the output feeds the next one. Stream elements (vertex or fragment arrays) are processed independently by kernels (vertex or fragment programs) and their output can be received again by another kernels.

The stream programming model allows an efficient computation, because kernels operate on independent elements from a set of input streams and can be processed using hardware like GPU, that process vertex or fragments streams in parallel. This allows making parallel computing without the complexity of traditional parallel programming models.

Computational Resources on GPU

In order to implement any kind of algorithm on GPU, there are different computational resources (Harris, 2005; Owens, 2007). By one side, current GPUs have two different parallel programmable processors: vertex and fragment processors. Vertex processors compute vertex streams (points with associated properties like position, color, normal, etc.). A vertex processor applies a vertex program to transform each input vertex to its position on the screen. Fragment processors compute fragment streams. They apply a fragment program to each fragment to calculate the final color of the pixel. In addition of using the attributes of each fragment, those processors can access to other data streams like textures when they are generating each pixel. Textures can be seen as an interface to access to read-only memory.

Another available resource in GPU is the rasterizer. It generates fragments using triangles built in from vertex and connectivity information. The rasterizer

allows generating an output set of data from a smaller input one, because it interpolates the properties of each vertex that belongs to a triangle (like color, texture coordinates, etc.) for each generated fragment.

One of the essential features of GPUs is the *render-to-texture* one. This allows storing the pixels generated by the fragments processor in a texture, instead of a screen buffer. This is at the moment the only mechanism to obtain directly output data from GPU computing. Render-to-texture cannot be thought as an interface to read-write memory, due to the fact that fragment processor can read data from a texture in multiple times, but it can write there just one time, at the end of each fragment processing.

ARTIFICIAL INTELLIGENCE ALGORITHMS ON GPU

Using the stream programming model as well as resources provided by graphics hardware, Artificial Intelligence algorithms can be parallelized and therefore computing-accelerated. The parallel and high-intensive computing nature of this kind of algorithms makes them good candidates for being implemented on the GPU.

Consider the evolution process of genetic algorithms, where a fitness value needs to be computed for each individual. Population could be considered as a data stream and fitness function as a kernel to process this stream. On GPU, for instance, the data stream must be represented as a texture, whereas the kernel must be implemented on a fragment program. Each individual's fitness would be obtained in an output stream, represented also by a texture, and obtained by the use of render-to-texture feature.

Recently some works have been realized mainly in paralleling ANN and EC algorithms, described in following sections.

Artificial Neural Networks

Bohn (1998) used GPGPU to reduce training time in Kohonen's feature maps. In this case, the bigger the map, the higher was the time reduction using the GPU. On 128x128 sized maps, time was similar using CPU and GPU, but on 512x512 sized maps, GPU was almost 3.5 times faster than CPU, increasing to 5.3 faster rates on 1024x1024 maps. This was one of the first implementations of GPGPU, made on a non-

programmable graphics system, a SiliconGraphics Infinite Reality workstation.

Later, with programmable hardware, Oh (2004) used the GPU for accelerating the process of obtaining the output of a multilayer perceptron ANN. Developed system was applied to pattern recognition obtaining 20x lower computing time than CPU implementation..

Considering another kind of ANNs, Zhongwen (2005) used GPGPU to reduce computing time in training Self-Organizing Maps (SOMs). The bigger the SOM, the higher was the reduction. Whereas using 128x128 neurons maps computing time was similar between CPU and GPU, 512x512 neuron maps involved a training process 4x faster using GPU implementation.

Bernhard (2005) used GPU to simulate Spiking Neurons model. This ANN model both requires high intensive calculations and has a parallel nature, so fits very well on GPGPU computation. Authors made different implementations depending on the neural network application. In the first case, an image segmentation algorithm was implemented using a locally-excitatory globally-inhibitory Spiking Neural Network (SNN). In this experiment, authors obtained up to 10x faster results. In the second case, SNNs were used to image segmentation using an algorithm based on histogram clustering where the ANN minimized the objective function. Here the speed was improved up to 10 times also.

Seoane (2007) showed multilayer perceptron ANN training time acceleration using GA. GPGPU techniques for ANN computing allowed accelerating it up to 11 times.

The company Evolved Machines (*Evolved Machines Website*) uses the powerful performance of GPUs to simulating of neural computation, obtaining results up to 100x faster than CPU computation.

Evolutionary Computation

In EC related works, Yu (2005) describes how parallel genetic algorithms can be mapped in low-cost graphics hardware. In their approach, chromosomes and fitness values are stored in textures. Fitness calculation and genetic operators were implemented using fragment programs on GPU. Different population sizes applied to the Colville minimization problem were used for testing, resulting in better time reductions according to bigger populations. In the case of a 128x128 sized population,

GPU genetic operators computing was 11.8 times faster than CPU, whereas in a 512x512 sized population, that rate incremented to 20.1. In fitness function computing, rates were 7.9 and 17.1 respectively.

In another work, Wong (2006) implemented Hybrid Genetic Algorithms on GPU incorporating the Cauchy mutation operator. All algorithm steps were implemented in graphics hardware, except random number generation. In this approach, a pseudo-deterministic method was proposed for selecting process, allowing significant running-time reductions. GPU implementation was 3x faster than CPU's one.

Fok (2007) showed how to implement evolutionary algorithms on GPU. Since the crossover operators of GA requires more complex calculations than mutation ones, authors studied a GPU implementation of Evolutionary Programming, using only mutation operators. Tests have been proved with the Cauchy distribution to 5 different optimization problems, obtaining between 1.25 and 5 times faster results.

FUTURE TRENDS

Nowadays GPUs are very powerful and they are evolving quite fast. By one side, there are more and more programmable elements in GPUs; by the other one, programming languages are becoming full-featured. There are more and more implementations of different kinds of general-purpose algorithms that take advantage of these features.

In Artificial Intelligence field the number of developments is rather low, in spite of the great amount of current algorithms and their high computing requirements. It seems very interesting using GPUs to extend existent implementations. For instance, some examples of speeding ANNs simulations up have been shown, however there is no works in accelerating training times. Likewise same ideas can be applied to implement other kinds of ANNs architectures or IA techniques, like in genetic programming field, where there is neither any development.

CONCLUSION

This paper has introduced general-purpose programming on GPUs. They have been shown as powerful parallel processors, which programming capabilities allow using for general-purpose high-intensive computing algorithms. Based on this idea, existent implementations of IA models like ANN or EC on GPUs have been described, with a considerable computing time reduction.

General-purpose computing on GPU and its use to accelerating IA algorithms provides great advantages, being an essential contribution in application where computing time is a decisive factor.

REFERENCES

Akenine-Möller, T. & Haines, E. (2002). *Real-Time Rendering.* Second Edition. A.K. Peters.

Bernhard, F. & Keriven, R. (2006). *Spiking Neurons on GPUs.* International Conference on Computational Science – ICCS 2006. 236-243.

Bohn, C.-A. (1998). *Kohonen Feature Mapping through Graphics Hardware.* In 3rd International Conference on Computational Intelligence and Neurosciences.

Buck, I. & Purcell, T. (2004). *A toolkit for computation on GPUs.* In GPU Gems. R. Fernando, editor. Addison-Wesley, 621-636.

Evolved Machines Website. (n.d.). Retrieved June 4, 2007 from http://www.evolvedmachines.com/

Fok, K.L., Wong, T.T. & Wong, M.L. (2007). *Evolutionary Computing on Consumer-Level Graphics Hardware.* IEEE Intelligent Systems. 22(2), 69-78.

General-Purpose Computation Using Graphics Hardware Website. (n.d.). Retrieved June 4, 2007 from http://www.gpu.org

Harris, M. (2005). *Mapping computational concepts to GPUs.* In GPU Gems 2, M. Pharr, editor. Addison-Wesley, 493-508.

HLSL Shaders. (n.d.). Retrieved June 4, 2007 from http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/directx9_c_Dec_2005/HLSL_Shaders.asp

Horn, D. (2005). *Stream reduction operations for GP-GPU applications.* In GPU Gems 2, M. Pharr, editor. Addison-Wesley, 573-589.

I

Lefohn, A., Kniss, J., Owens J. (2005). *Implementing efficient parallel data structures on GPUs.* In GPU Gems. 2, M. Pharr, editor. Addison-Wesley, 521-545.

Luebke, D. (2006). *General-Purpose Computation on Graphics Hardware.* In Supercomputing 2006 Tutorial on GPGPU.

Mark, W.R., Glanville, R.S., Akeley, K. & Kilgard, M.J. (2003). *Cg: a system for programming graphics hardware in a C-like language.* ACM Trans. Graph. ACM Press, 22(3), 896-907.

Oh, K.-S., Jung, K. (2004). *GPU implementation of neural networks.* Pattern Recognition. 37(6), 1311-1314.

OpenGL Shading Language Information Site. (n.d.). Retrieved June 4, 2007, from http://developer.3dlabs.com/openGL2/index.htm

Owens, J. (2005a). *Streaming architectures and technology trends.* In GPU Gems 2, M. Pharr, editor. Addison-Wesley, 457-470.

Owens, J. (2005b). *The GPGPU Progamming Model.* In General Purpose Computation on Graphics Hardware. IEEE Visualization 2005 Tutorial.

Owens, L.D., Luebke, D., Govindaraju, N., Harris, M., Kruger, J., Lefohn, A.E. & Purcell, T.J. (2007). *A Survey of General-Purpose Computation on Graphics Hardware.* COMPUTER GRAPHICS forum. 26(1), 80-113.

Seoane, A., Rabuñal, J. R. & Pazos, A. (2007). *Aceleración del Entrenamiento de Redes de Neuronas Artificiales mediante Algoritmos Genéticos utilizando la GPU.* Actas del V Congreso Español sobre Meta-heurísticas, Algorítmos Evolutivos y Bioinspirados - MAEB 2007. 69-76.

Wong, M.L. & Wong, T.T. (2006). *Parallel Hybrid Genetic Algorithms on Consumer-Level Graphics Hardware.* IEEE Congress on Evolutionary Computation. 2973-2980.

Yu, Q., Chen, C. & Pan, Z. (2005). *Parallel Genetic Algorithms on Programmable Graphics Hardware.* Lecture Notes in Computer Science. 1051-1059.

Zhongwen, L., Hongzhi, L., Zhengping, Y. & Xincai, W. (2005). *Self-Organizing Maps Computing on Graphic Process Unit.* ESANN'2005 proceedings - European Symposium on Artificial Neural Networks. 557-562.

KEY TERMS

Fragment: Potential pixel containing all the necessary information (color, depth, etc.) to generate the final fragment color.

Fragment Processor: Graphics system element that receives as input a set of fragments and processes it to obtain pixel, writing them in a target buffer. Present GPUs have multiple fragment processors working in parallel and can be programmed using fragment programs.

Graphics Pipeline: Three dimensional graphics oriented architecture, composed by several stages that run sequentially.

Graphics Processing Unit (GPU): Electronic device designed for graphics rendering in computers. Its architecture is specialized in graphics calculations.

General-Purpose Computation on GPUs (GP-GPU): Trend in computing devices dedicated to implement general-purpose algorithms using graphics devices, called GPUs. At the moment, the high programmability and performance of GPUs allow developers run classical algorithms in these devices to speed non-graphics applications up, especially those algorithms with parallel nature.

Pixel: Picture Element abbreviation, used for referring graphic image points.

Rasterizer: Graphics Pipeline element, which from graphic primitives provides appropriate fragments to a target buffer.

Render-to-Texture: GPU feature that allows stocking the fragment processor output on a texture instead on a screen buffer.

Stream Programming Model: This parallel programming model is based on defining, by one side, sets of input and output data, called streams, and by the other side, intensive computing operations, called

kernel functions, to be applied sequentially on the streams.

Texture: In computer graphics field, it refers to a digital image used to modify the appearance of a tri-dimensional object. The operation that wraps around a texture over an object is called texture mapping. Talking about GPGPU, a texture can be considered as a data stream.

Vertex: In computer graphics field, it refers to a clearly defined point in a tridimensional space, which is processed by Graphics Pipeline. Relationships can be established between those vertices (like triangles) to assembly structures that define a tridimensional object. Talking about GPGPU, an vertex array can be considered as a data stream.

Vertex Processor: Graphics system component that receives as input a set of 3D vertex and process them to obtain 2D screen positions. Present GPUs have multiple vertex processors working in parallel and can be programmed using vertex programs.

Improving the Naïve Bayes Classifier

Liwei Fan
National University of Singapore, Singapore

Kim Leng Poh
National University of Singapore, Singapore

INTRODUCTION

A **Bayesian Network** (BN) takes a relationship between graphs and probability distributions. In the past, BN was mainly used for knowledge representation and reasoning. Recent years have seen numerous successful applications of BN in classification, among which the Naïve Bayes classifier was found to be surprisingly effective in spite of its simple mechanism (Langley, Iba & Thompson, 1992). It is built upon the strong assumption that different attributes are independent with each other. Despite of its many advantages, a major limitation of using the Naïve Bayes classifier is that the real-world data may not always satisfy the independence assumption among attributes. This strong assumption could make the prediction accuracy of the Naïve Bayes classifier highly sensitive to the correlated attributes. To overcome the limitation, many approaches have been developed to improve the performance of the Naïve Bayes classifier.

This article gives a brief introduction to the approaches which attempt to relax the independence assumption among attributes or use certain pre-processing procedures to make the attributes as independent with each other as possible. Previous theoretical and empirical results have shown that the performance of the Naïve Bayes classifier can be improved significantly by using these approaches, while the computational complexity will also increase to a certain extent.

BACKGROUND

The **Naïve Bayes classifier**, also called simple Bayesian classifier, is essentially a simple BN. Since no **structure learning** is required, it is very easy to construct and implement a Naïve Bayes classifier. Despite its simplicity, the Naïve Bayes classifier is competitive with other more advanced and sophisticated classifiers such as

decision trees (Friedman, Geiger & Goldszmidt, 1997). Owing to these advantages, the Naïve Bayes classifier has gained great popularity in solving different classification problems. Nevertheless, its **independence assumption** among attributes is often violated in the real world. Fortunately, many approaches have been developed to alleviate this problem.

In general, these approaches can be divided into two groups. One attempts to relax the independence assumption of Naïve Bayes classifier, e.g. Semi-Naïve Bayes (SNB) (Kononenko, 1991), Searching for dependencies (Pazzani, 1995), the Tree Augmented Naïve Bayes (TAN) (Friedman, Geiger & Goldszmidt, 1997), SuperParent Tree Augmented Naïve Bayes (SP-TAN) (Keogh & Pazzani, 1999), Lazy Bayes Rule (LBR) (Zheng & Webb, 2000) and Aggregating One-Dependence Estimators (AODE) (Webb, Boughton & Wang, 2005).

The other group attempts to use certain **pre-processing procedures** to select or transform the attributes, which can be more suitable for the assumption of the Naïve Bayes classifier. The Feature selection can be implemented by greedy forward search (Langley & Sage, 1994) and **Decision Trees** (Ratanamahatana & Gunopulos, 2002). The transformation techniques include **Principal Component Analysis** (PCA) (Gupta, 2004), **Independent Component Analysis** (ICA) (Prasad, 2004) and CC-ICA (Bressan & Vitria, 2002). The next section describes the main ideas of the two groups of techniques in a broad way.

IMPROVING THE NAÏVE BAYES CLASSIFIER

This section introduces the two groups of approaches that have been used to improve the **Naïve Bayes classifier**. In the first group, the strong independence assumption is relaxed by restricted structure learning. The second

group helps to select some major (and approximately independent) attributes from the original attributes or transform them into some new attributes, which can then be used by the Naïve Bayes classifier.

Relaxing the Independence Assumption

Relaxing the **independence assumption** means that the dependence will be considered in constructing the network. To consider the dependencies between attributes, Kononenko (Kononenko, 1991) proposed the Semi-Naïve Bayes classifier (SNB), which joined the attributes based on the theorem of Chebyshev. The medical diagnostic data were used to compare the performance of the SNB and the NB. It was found that the results of two domains are identical but in the other two domains SNB slightly improves the performance. Nevertheless, this method may cause overfitting problems. Another limitation of the SNB is that the number of parameters will grow exponentially with the increase of the number of attributes that need to be joined. In addition, the exhaustive searching technique of joining attributes may affect the computational time. Pazzani (Pazzani, 1995) used Forward Sequential Selection and Joining (FSSJ) and Backward Sequential Elimination and Joining (BSEJ) to search dependencies and join the attributes. They tested the two methods on UCI data and found that BSEJ provided the most improvement.

Friedman et al. (Friedman, Geiger & Goldszmidt, 1997) found that Kononenko's and Pazzani's methods can be represented as an augmented Naïve Bayes network, which includes some subgraphs. They restricted the network to be Tree Augmented Naïve Bayes (TAN) that spans over all attributes and can be learned by tree-structure learning algorithms. The results based on problems from the UCI repository showed that the TAN classifier outperforms the Naïve Bayes classifier. It is also competitive with C4.5 while maintains the computational simplicity. However, the use of the TAN classifier is only limited to the problems with discrete attributes. For the problems with continuous attributes, these attributes must be prediscretized. To address this problem, Friedman et al. (Friedman, Goldszmidt & Lee, 1998) extended TAN to deal with continuous attributes via parametric and semiparametric conditional probabilities. Keogh & Pazzani (Keogh & Pazzani, 1999) proposed a variant of the TAN classifier, i.e. SP-TAN, which could result in better performance than TAN. The performance of SP-TAN is also competitive with the Lazy Bayes Rule (LBR), in which the lazy learning techniques are used in the Naïve Byes classifier (Zheng, & Webb, 2000; Wang & Webb, 2002)

Although LBR and SP-TAN have outstanding performance on the testing data, the main disadvantage of the two methods is that they have high computational complexity. Aggregating One-Dependence Estimators (AODE), developed by Webb et al. (Webb, Boughton & Wang, 2005), can avoid model selection which may reduce computational complexity and lead to lower variance. These advantages have been demonstrated by some empirical experiment results. It is also empirically found that the average prediction accuracy of AODE is comparative to that of LBR and SP-TAN but with lower variance. Therefore, AODE might be more suitable for small datasets due to its lower variance.

Using Pre-Processing Procedures

In general, the **pre-processing procedures** for the Naïve Bayes classifier include feature selection and transforming the original attributes. The Selective Bayes classifier (SBC) (Langley & Sage, 1994) deals with correlated features by selecting only some attributes into the final classifier. They used a greedy method to search the space and forward selection to select the attributes. In their study, six UCI datasets are used to compare the performance of the Naïve Bayes classifier, SBC and C4.5. It is found that selecting the attributes can improve the performance of the Naïve Bayes classifier when there are redundant attributes. In addition, SBC is found to be competitive with C4.5 in terms of the datasets by which C4.5 outperforms the Naïve Bayes classifier. The study by Ratanamahatana & Gunopulos (Ratanamahatana & Gunopulos, 2002) applied C4.5 to select the attributes for the Naïve Bayes classifier. Interestingly, experimental results showed that the new attributes obtained by C4.5 can make the Naïve Bayes classifier outperform C4.5 with respect to a number of datasets.

Transforming the attributes is another useful pre-processing procedure for the Naïve Bayes classifier. Gupta (Gupta, 2004) found that **Principal Component Analysis** (PCA) was helpful to improve the classification accuracy and reduce the computational complexity. Prasad (Prasad, 2004) applied **Independent Component Analysis** (ICA) to all the training data and found

that the performance of Naïve Bayes classifier integrated ICA performed better than C4.5 and IB1 integrated with ICA. Bressan and Vitria (Bressan & Vitria, 2002) proposed the class-conditional ICA (CC-ICA) to do pre-processing procedure for the Naïve Bayes classifier, and found that CC-ICA based Naïve Bayes classifier outperformed the pure Naïve Bayes classifier.

Based on the UCI datasets, a detailed comparative study of PCA, ICA and CC-ICA for Naïve Bayes classifier has been carried out by Fan & Poh (Fan & Poh, 2007). PCA attempts to transform the original data into a new uncorrelated dataset, while ICA attempts to transform them into a new dataset with independent attributes. Class-conditional ICA (CC-ICA), proposed by Bressan and Vitria (2002), is built upon the idea that ICA is used to make the attributes as independent as possible for each class. In such a way, the new attributes are more reasonable than those from the PCA and ICA in order to satisfy the independence assumption of the Naïve Bayes classifier.

The datasets were limited to the continuous datasets due to the requirement of the three pre-processing procedures. The results showed that all the three pre-processing procedures can improve the performance of the Naïve Bayes classifier. It is likely due to the fact that transforming the attributes could weaken the dependence among different attributes. In addition, the discrepancy between the performance of ICA and PCA integrated with the Naïve Bayes classifier is not large. This may be an indication that PCA and ICA are competitive in improving the performance of Naïve Bayes classifier. When the number of attributes became larger, the three pre-processing procedures also improved the performance of the Naïve Bayes classifier by more.

From the methodological point of view, the CC-ICA pre-processing procedure seems to be more plausible than PCA and ICA for Naïve Bayes classifier (Bressan and Vitria, 2002; Vitria, Bressan, & Radeva, 2007). The experimental results by Fan & Poh (Fan & Poh, 2007) also showed that CC-ICA integrated with the Naïve Bayes classifier outperforms PCA and ICA integrated with the Naïve Bayes classifier in terms of classification accuracy. However, CC-ICA requires more training data to ensure that there are enough training data for each class. It is therefore suggested that the choice of a suitable pre-processing procedure should depend on the characteristics of datasets, e.g. the sample size for each class.

FUTURE TRENDS

With the development of the algorithms for learning BN, relaxing the independence assumption is promising for improving the performance of the Naïve Bayes classifier. However, relaxing the independence assumption to the unrestricted BN is not appropriate. Friedman et al. (Friedman, Geiger, & Goldszmidt, 1997) compared the Naïve Bayes classifier and Bayesian Network and found that using unrestricted BN did not improve the accuracy. On the contrary, it even reduced the accuracy in some domains. Therefore, other restricted BN may be used for improving the performance while keeping the simplicity of the Naïve Bayes classifier. Effective and simple learning algorithm is also important for the improving the performance.

On the other hand, with the development of algorithms for machine learning, more pre-processing procedures are expected to be developed for selecting or transforming the attributes. One possible way to get better performance is to combine feature selection with transformation techniques to do the pre-processing procedures. Among the alternative techniques for doing pre-processing procedures, the most promising one might be ICA. The reason is that the motivation of the pre-processing procedures is to derive the attributes satisfying the independence assumption for the Naïve Bayes classifier while the objective of ICA is to find the independent components. However, there are also some limitations on the use of ICA, e.g. the requirements of continuous datasets and a large number of training samples. How to overcome these limitations is therefore a potential area for future research.

CONCLUSION

This article briefly discusses the techniques which can be used to improve the performance of the Naïve Bayes classifier. The general idea is to overcome the limitation of the strong independence assumption of the **Naïve Bayes classifier**. Relaxing the strong assumption is a natural way and has been studied from different viewpoints. All the approaches relaxing the assumption discussed in the article is restricted **Bayesian Networks**, which are still most practicable techniques. In addition, pre-processing procedures are also very useful to make the attributes to satisfy the independence assumption. However, using these approaches increases the compu-

tational complexity to a certain extent. It would be useful to model correlations among appropriate attributes that can be captured by simple restricted structure but with good performance.

REFERENCES

Bressan, M., Vitria, J. (2002). Improving Naïve Bayes Using Class-conditional ICA. *Advances in Artificial Intelligence - IBERAMIA 2002.* 1-10.

Cheng, J., & Greiner, R. (1999). Comparing Bayesian Network Classifiers. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence.* 101-107.

Fan, L., Poh, K.L. (2007). A Comparative Study of PCA, ICA and Class-conditional ICA for Naïve Bayes Classifier. *Computational and Ambient Intelligence. Lecture Notes in Computer Science.* (4507), 16-22.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning.* (29) 131-163.

Friedman, N., Goldszmidt, M., & Lee, T.J. (1998). Bayesian Network Classification with Continuous Attributes: Getting the best of Both Discretization and Parametric Fitting. *Proceedings of the Fifteenth International Conference on Machine Learning.* 179-187.

Gupta, G. K. (2004). Principal Component Analysis and Bayesian Classifier Based on Character Recognition. *Bayesian Inference and Maximum Entropy Methods in Science and Engineering. AIP Conference Proceedings.* (707), 465-479.

Keogh, E., & Pazzani, M.J. (1999). Learning Augmented Bayesian Classifiers: A Comparison of Distribution-based and Classification-based Approaches. *Proceedings of the International Workshop on Artificial Interlligence and Statistics.* 225-230.

Kononenko, I. (1991). Semi-Naïve Bayesian Classifier. *Proceedings of the sixth Eurjopean Working Session on Learning.* 206-219.

Langley, P., Iba, W., & Thompson, K. (1992). An Analysis of Bayesian Classifiers. *Proceedings of the Tenth National Conference on Artificial Intelligence.* AAAI Press, San Jose, CA. 223-228.

Langley, P. & Sage, S. (1994). Induction of Selective Bayesian Classifiers. *Proceedings of the Tenth Conference on Unvertainty in Artificial Intelligence.* 399-406.

Pazzani, M.J. (1995). Searching for Dependencies in Bayesian Classifiers. *Proceedings of the fifth International Workshop on Artificial Intelligence and Statistics.* 424-429.

Prasad, M.N., Sowmya, A., Koch, I. (2004). Feature Subset Selection using ICA for Classifying Emphysema in HRCT Images. *Proceedings of the 17th International Conference on Pattern Recognition.* 515-518.

Ratanamahatana, C.A. and Gunopulos, D., Feature selection for the naive Bayesian classifier using decision trees. *Applied Artificial Intelligence.* (17), 475-487.

Vitria, J., Bressan, M., & Radeva, P. (2007). Bayesian Classification of Cork Stoppers Using Class-conditional Independent Component Analysis. *IEEE Transactions on Systems, Man and Cybernetics.* (37), 32-38.

Wang, Z., & Webb, G.I. (2002). Comparison of Lazy Bayesian Rule and Tree-Augmented Bayesian Learning. *Proceedings of the IEEE International Conference on Data Mining.* 775-778.

Webb, G., Boughton, J.R., & Wang, Z. (2005). Not so Naïve Bayes: Aggregating One-Dependence Estimatiors. *Machine Learning.* (58), 5-24.

Zheng, Z., & Webb, G. (2000). Lazy Learning of Bayesian Rules. *Machine Learning.* (41), 53-87.

KEY TERMS

Decision Trees: Decision tree is a classifier in the form of a tree structure, where each node is either a leaf node or a decision node. A decision tree can be used to classify an instance by starting at the root of the tree and moving through it until a leaf node, which provides the classification of the instance. A well known and frequently used algorithm of decision tree over the years is C4.5.

Forward Selection and Backward Elimination: A forward selection method would start with the empty set and successively add attributes, while a backward

elimination process would begin with the full set and remove unwanted ones.

Greedy Search: At each point in the search, the algorithm considers all local changes to the current set of attributes, makes its best selection, and never reconsiders this choice.

Independent Component Analysis (ICA): Independent component analysis (ICA) is a newly developed technique for finding hidden factors or components to give a new representation of multivariate data. ICA could be thought of as a generalization of PCA. PCA tries to find uncorrelated variables to represent the original multivariate data, whereas ICA attempts to obtain statistically independent variables to represent the original multivariate data.

Naïve Bayes Classifier: The Naïve Bayes classifier, also called simple Bayesian classifier, is essentially a simple Bayesian Network (BN). There exist two underlying assumptions in the Naïve Bayes classifier. First, all attributes are independent with each other given the classification variable. Second, all attributes are directly dependent on the classification variable. Naïve Bayes classifier computes the posterior of classification variable given a set of attributes by using the Bayes rule under the conditional independence assumption.

Principal Component Analysis (PCA): PCA is a popular tool for multivariate data analysis, feature extraction and data compression. Given a set of multivariate measurements, the purpose of PCA is to find a set of variables with less redundancy. The redundancy is measured by correlations between data elements.

UCI Repository: This is a repository of databases, domain theories and data generator that are used by the machine learning community for the empirical analysis of machine learning algorithms.

Incorporating Fuzzy Logic in Data Mining Tasks

Lior Rokach
Ben Gurion University, Israel

In this chapter we discuss how fuzzy logic extends the envelop of the main data mining tasks: clustering, classification, regression and association rules. We begin by presenting a formulation of the data mining using fuzzy logic attributes. Then, for each task, we provide a survey of the main algorithms and a detailed description (i.e. pseudo-code) of the most popular algorithms.

INTRODUCTION

There are two main types of uncertainty in supervised learning: statistical and cognitive. Statistical uncertainty deals with the random behavior of nature and all existing data mining techniques can handle the uncertainty that arises (or is assumed to arise) in the natural world from statistical variations or randomness. Cognitive uncertainty, on the other hand, deals with human cognition.

Fuzzy set theory, first introduced by Zadeh in 1965, deals with cognitive uncertainty and seeks to overcome many of the problems found in classical set theory. For example, a major problem faced by researchers of control theory is that a small change in input results in a major change in output. This throws the whole control system into an unstable state. In addition there was also the problem that the representation of subjective knowledge was artificial and inaccurate. Fuzzy set theory is an attempt to confront these difficulties and in this chapter we show how it can be used in data mining tasks.

BACKGROUND

Data mining is a term coined to describe the process of sifting through large and complex databases for identifying valid, novel, useful, and understandable patterns and relationships. Data mining involves the inferring of algorithms that explore the data, develop the model and discover previously unknown patterns. The model is used for understanding phenomena from the data, analysis and prediction. The accessibility and abundance of data today makes knowledge discovery and data mining a matter of considerable importance and necessity.

We begin by presenting some of the basic concepts of fuzzy logic. The main focus, however, is on those concepts used in the induction process when dealing with data mining. Since fuzzy set theory and fuzzy logic are much broader than the narrow perspective presented here, the interested reader is encouraged to read Zimmermann (2005).

In classical set theory, a certain element either belongs or does not belong to a set. Fuzzy set theory, on the other hand, permits the gradual assessment of the membership of elements in relation to a set.

Let U be a universe of discourse, representing a collection of objects denoted generically by u. A fuzzy set A in a universe of discourse U is characterized by a membership function μ_A which takes values in the interval $[0, 1]$. Where $\mu_A(u) = 0$ means that u is definitely not a member of A and $\mu_A(u) = 1$ means that u is definitely a member of A.

The above definition can be illustrated on the vague set of *Young*. In this case the set U is the set of people. To each person in U, we define the degree of membership to the fuzzy set *Young*. The membership function answers the question "to what degree is person u young?". The easiest way to do this is with a membership function based on the person's age. For example Figure 1 presents the following membership function:

$$\mu_{Young}(u) = \begin{cases} 0 & age(u) > 32 \\ 1 & age(u) < 16 \\ \dfrac{32 - age(u)}{16} & otherwise \end{cases}$$

(1)

Figure 1. Membership function for the young set
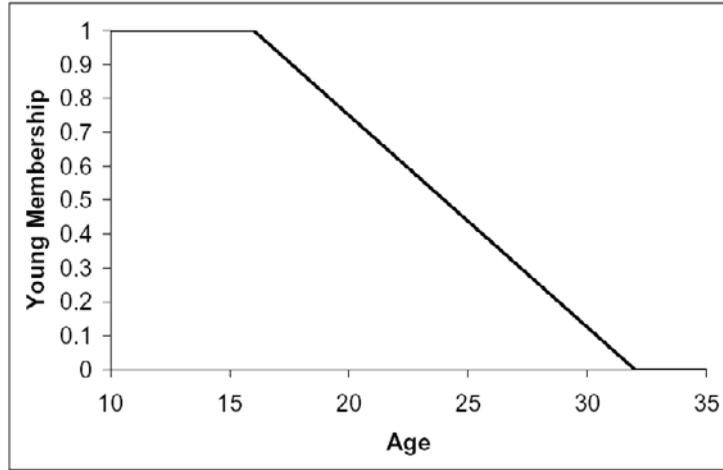
Given this definition, John, who is 18 years old, has degree of youth of 0.875. Philip, 20 years old, has degree of youth of 0.75. Unlike probability theory, degrees of membership do not have to add up to 1 across all objects and therefore either many or few objects in the set may have high membership. However, an object's membership in a set (such as "young") and the set's complement ("not young") must still sum to 1.

The main difference between classical set theory and fuzzy set theory is that the latter admits to partial set membership. A classical or crisp set, then, is a fuzzy set that restricts its membership values to {0,1}, the

Figure 2. Membership function for the crisp young set
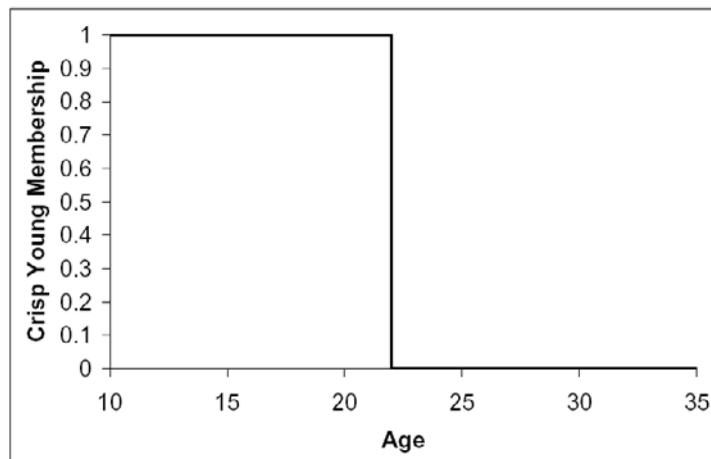
endpoints of the unit interval. Membership functions can be used to represent a crisp set. For example, Figure 2 presents a crisp membership function defined as:

$$\mu_{CrispYoung}(u) = \begin{cases} 0 & age(u) > 22 \\ 1 & age(u) \leq 22 \end{cases}$$

(2)

In regular classification problems, we assume that each instance takes one value for each attribute and that each instance is classified into only one of the mutually exclusive classes. To illustrate how fuzzy logic can help data mining tasks, we introduce the problem of modeling the preferences of TV viewers. In this problem there are 3 input attributes: A = {Time of Day,Age Group,Mood}. The classification can be the movie genre that the viewer would like to watch, such as C = {Action,Comedy,Drama}. All the attributes are vague by definition. For example, people's feelings of happiness, indifference, sadness, sourness and grumpiness are vague without any crisp boundaries between them. Although the vagueness of "Age Group" or "Time of Day" can be avoided by indicating the exact age or exact time, a rule induced with a crisp decision tree may then have an artificial crisp boundary, such as "IF Age < 16 THEN action movie". But how about someone who is 17 years of age? Should this viewer definitely not watch an action movie? The viewer preferred genre may still be vague. For example, the viewer may be in a mood for both comedy and drama movies. Moreover, the association of movies into genres may also be vague. For instance the movie "Lethal Weapon" (starring Mel Gibson and Danny Glover) is considered to be both comedy and action movie.

Fuzzy concept can be introduced into a classical data mining task if at least one of the attributes is fuzzy. In the example described above , both input and target attributes are fuzzy. Formally the problem is defined as following: Each class c_j is defined as a fuzzy set on the universe of objects U. The membership function $\mu_{cj}(u)$ indicates the degree to which object u belongs to class c_j. Each attribute a_i is defined as a linguistic attribute which takes linguistic values from $dom(a_i)$ = {$v_{i,1}$, $v_{i,2}$,...$v_{i,|dom(aj)|}$}. Each linguistic value $v_{i,k}$ is also a fuzzy set defined on U. The membership $\mu_{v_{i,k}}(u)$ specifies the degree to which object u's attribute a_i is $v_{i,k}$. Recall that the membership of a linguistic value can be subjectively assigned or transferred from numerical values by a membership function defined on the range of the numerical value.

Typically, before one can incorporate fuzzy concepts into a data mining application, an expert is required to provide the fuzzy sets for the quantitative attributes, along with their corresponding membership functions (Mitra and Pal, 2005). Alternatively the appropriate fuzzy sets are determined using fuzzy clustering.

MAIN FOCUS OF THE CHAPTER

Fuzzy Supervised Learning

In this section we survey supervised methods that incorporate fuzzy sets. Supervised methods are methods that attempt to discover the relationship between input attributes and a target attribute (sometimes referred to as a dependent variable). The relationship discovered is represented in a structure referred to as a model. Usually models describe and explain phenomena, which are hidden in the dataset and can be used for predicting the value of the target attribute knowing the values of the input attributes.

It is useful to distinguish between two main supervised models: classification models (classifiers) and Regression Models. Regression models map the input space into a real-value domain. For instance, a regressor can predict the demand for a certain product given its characteristics. On the other hand, classifiers map the input space into pre-defined classes.

Fuzzy set theoretic concepts can be incorporated at the input, output, or into to backbone of the classifier. The data can be presented in fuzzy terms and the output decision may be provided as fuzzy membership values (Peng, 2004). In this chapter we will concentrate on fuzzy decision trees. The interested reader is encouraged to read also about soft regression (Shnaider et al., 1997) and Neuro-fuzzy (Mitra and Hayashi, 2000, Nauck, 1997).

Decision tree is a predictive model which can be used to represent classifiers. Decision trees are frequently used in applied fields such as finance, marketing, engineering and medicine. Decision tree are self-explained. There is no need to be an expert in data mining in order to follow a certain decision tree.

There are several algorithms for induction of fuzzy decision trees (Olaru and Wehenkel, 2003), most of them extend existing decision trees methods such as: Fuzzy-CART (Jang, 1994), Fuzzy-ID3 (Cios and Sztandera, 1992; Maher and Clair, 1993). Another complete

framework for building a fuzzy tree including several inference procedures based on conflict resolution in rule-based systems and efficient approximate reasoning methods was presented in (Janikow, 1998).

In this section we will focus on the algorithm proposed in Yuan and Shaw (1995). This algorithm can handle the classification problems with both fuzzy attributes and fuzzy classes represented in linguistic fuzzy terms. It can also handle other situations in a uniform way where numerical values can be fuzzified to fuzzy terms and crisp categories can be treated as a special case of fuzzy terms with zero fuzziness. The algorithm uses classification ambiguity as fuzzy entropy. The classification ambiguity directly measures the quality of classification rules at the decision node. It can be calculated under fuzzy partitioning and multiple fuzzy classes.

When a certain attribute is numerical, it needs to be fuzzified into linguistic terms before it can be used in the algorithm (Hong et al., 1999). The fuzzification process can be performed manually by experts or can be derived automatically using some sort of clustering algorithm. Clustering groups the data instances into subsets in such a manner that similar instances are grouped together; different instances belong to different groups. The instances are thereby organized into an efficient representation that characterizes the population being sampled.

One can use a simple algorithm to generate a set of membership functions on numerical data. Assume attribute a_i has numerical value x from the domain X. We can cluster X to k linguistic terms $v_{i,j}, j = 1,...,k$. The size of k is manually predefined. Figure 3 illustrates the creation of four groups defined on the age attribute: "young", "early adulthood", "middle-aged" and "old age". Note that the first set ("young") and the last set ("old age") have a trapezoidal form which can be uniquely described by the four corners. For example, the "young" set could be represented as (0,0,16,32). In between, all other sets ("early adulthood" and "middle-aged") have a triangular form which can be uniquely described by the three corners. For example, the set "early adulthood" is represented as (16,32,48).

The induction algorithm of fuzzy decision tree measures the classification ambiguity associated with each attribute and split the data using the attribute with the smallest classification ambiguity. The classification ambiguity of attribute a_i with linguistic terms $v_{i,j}$, $j = 1,...,k$ on fuzzy evidence S, denoted as $G(a_i \mid S)$, is the weighted average of classification ambiguity calculated as:

$$G(a_i \mid S) = \sum_{j=`1}^{k} w(v_{i,j} \mid S) \cdot G(v_{i,j} \mid S)$$

(3)

Figure 3. Membership function for various groups in the age attribute
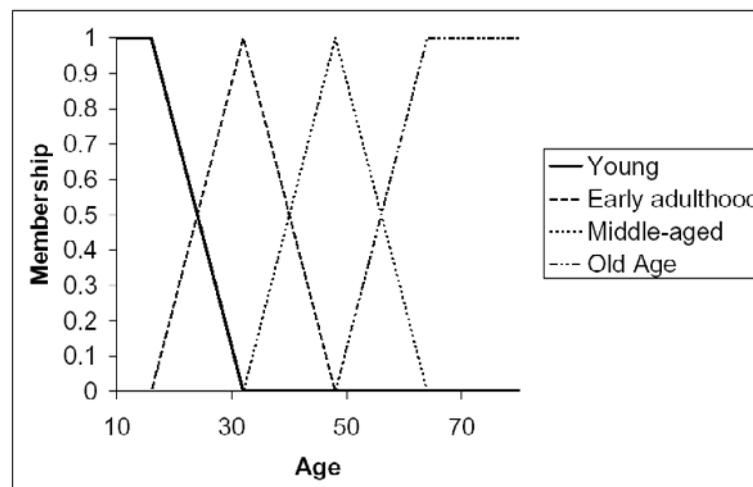
where $w(v_{i,j} \mid S)$ is the weight which represents the relative size of $v_{i,j}$ and is defined as:

$$w(v_{i,j} \mid S) = \frac{M(v_{i,j} \mid S)}{\sum_k M(v_{i,k} \mid S)} \qquad (4)$$

The classification ambiguity of $v_{i,j}$ is defined as

$$G(v_{i,j} \mid S) = g\left(\vec{p}\left(C \mid v_{i,j} \right) \right),$$

which is measured based on the possibility distribution vector

$$\vec{p}\left(C \mid v_{i,j} \right) = \left(p\left(c_1 \mid v_{i,j} \right), ..., p\left(c_{|k|} \mid v_{i,j} \right) \right).$$

Given $v_{i,j}$, the possibility of classifying an object to class c_l can be defined as:

$$p\left(c_l \mid v_{i,j} \right) = \frac{S(v_{i,j}, c_l)}{\max_k S(v_{i,j}, c_k)} \qquad (5)$$

where $S(A,B)$ is the fuzzy subsethood that measures the degree to which A is a subset of B. The subsethood can be used to measure the truth level of the rule of classification rules. For example given a classification rule such as "IF Age is Young AND Mood is Happy THEN Comedy" we have to calculate $S(Hot \cap Sunny, Swimming)$ in order to measure the truth level of the classification rule.

The function $g\left(\vec{p} \right)$ is the possibilistic measure of ambiguity or nonspecificity and is defined as:

$$g\left(\vec{p} \right) = \sum_{i=1}^{|\vec{p}|} \left(p_i^* - p_{i+1}^* \right) \cdot \ln(i) \qquad (6)$$

where

$$\vec{p}^* = \left(p_1^*, ..., p_{|\vec{p}|}^* \right)$$

is the permutation of the possibility distribution $\vec{p}$ sorted such that $p_i^* \geq p_{i+1}^*$. All the above calculations are carried out at a predefined significant level α. An instance will take into consideration of a certain branch $v_{i,j}$ only if its corresponding membership is greater

than α. This parameter is used to filter out insignificant branches.

After partitioning the data using the attribute with the smallest classification ambiguity, the algorithm looks for nonempty branches. For each nonempty branch, the algorithm calculates the truth level of classifying all instances within the branch into each class. The truth level is calculated using the fuzzy subsethood measure $S(A,B)$.

If the truth level of one of the classes is above a predefined threshold β then no additional partitioning is needed and the node become a leaf in which all instance will be labeled to the class with the highest truth level. Otherwise the procedure continues in a recursive manner. Note that small values of β will lead to smaller trees with the risk of underfitting. A higher β may lead to a larger tree with higher classification accuracy. However, at a certain point, higher values β may lead to overfitting.

In a regular decision tree, only one path (rule) can be applied for every instance. In a fuzzy decision tree, several paths (rules) can be applied for one instance. In order to classify an unlabeled instance, the following steps should be performed:

- Step 1: Calculate the membership of the instance for the condition part of each path (rule). This membership will be associated with the label (class) of the path.
- Step 2: For each class calculate the maximum membership obtained from all applied rules.
- Step 3: An instance may be classified into several classes with different degrees based on the membership calculated in Step 2.

Fuzzy Clustering

The goal of clustering is descriptive, that of classification is predictive. Since the goal of clustering is to discover a new set of categories, the new groups are of interest in themselves, and their assessment is intrinsic. In classification tasks, however, an important part of the assessment is extrinsic, since the groups must reflect some reference set of classes.

Clustering groups data instances into subsets in such a manner that similar instances are grouped together, while different instances belong to different groups. The instances are thereby organized into an efficient representation that characterizes the population being

sampled. Formally, the clustering structure is represented as a set of subsets $C = C_1,...,C_k$ of S, such that:

$$S = \bigcup_{i=1}^{k} C_i$$

and $C_i \cap C_j = \emptyset$ for $i \neq j$. Consequently, any instance in S belongs to exactly one and only one subset.

Traditional clustering approaches generate partitions; in a partition, each instance belongs to one and only one cluster. Hence, the clusters in a hard clustering are disjointed. Fuzzy clustering (Nasraoui and Krishnapuram, 1997, Shnaider et al., 1997) extends this notion and suggests a *soft clustering* schema. In this case, each pattern is associated with every cluster using some sort of membership function, namely, each cluster is a fuzzy set of all the patterns. Larger membership values indicate higher confidence in the assignment of the pattern to the cluster. A hard clustering can be obtained from a fuzzy partition by using a threshold of the membership value.

The most popular fuzzy clustering algorithm is the fuzzy *c*-means (FCM) algorithm. FCM is an iterative algorithm. The aim of FCM is to find cluster centers (centroids) that minimize a dissimilarity function. To accommodate the introduction of fuzzy partitioning, the membership matrix(U) is randomly initialized according to Equation 7.

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1,...,n \tag{7}$$

The algorithm minimizes a dissimilarity (or distance) function which is given in Equation 13:

$$J(U,c_1,c_2,...,c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 \tag{8}$$

where, u_{ij} is between 0 and 1; c_i is the centroid of cluster i; d_{ij} is the Euclidian distance between i-th centroid and j-th data point; m is a weighting exponent.

To reach a minimum of dissimilarity function there are two conditions. These are given in Equation 9 and Equation 10.

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{9}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}} \tag{10}$$

By iteratively updating the cluster centers and the membership grades for each data point, FCM iteratively moves the cluster centers to the "right" location within a data set. However, FCM does not ensure that it converges to an optimal solution. The random initialization of U might have uncancelled effect on the final performance.

Fuzzy Association Rules

Association rules are rules of the kind "70% of the customers who buy vine and cheese also buy grapes". While the traditional field of application is market basket analysis, association rule mining has been applied to various fields since then, which has led to a number of important modifications and extensions.

A fuzzy association algorithm is proposed in Komem and Schneider (2005). The quantitative values are first transformed into a set of membership grades, by using predefined membership functions. Every membership grade represents the agreement of a quantitative value with a linguistic term. In order to avoid discriminating the importance level of data, each point must have membership grade of 1 in one membership function; Thus, the membership functions of each attribute produce a continuous line of $\mu = 1$. Additionally, in order to diagnose the bias direction of an item from the center of a membership function region, almost each point get another membership grade which is lower than 1 in other membership functions region. Thus, each end of membership function region is touching, close to, or slightly overlapping an end of another membership function (except the outside regions, of course).

By this mechanism, as point "a" moves right, further from the center of the region "middle", it gets a higher value of the label "middle-high", additionally to the value 1 of the label "middle".

FUTURE TRENDS

Some of the challenges of using fuzzy theory in data mining tasks, include the following:

1. Incorporation of domain knowledge for improving the fuzzy modeling.
2. Developing methods for presenting fuzzy data model to the end-users.
3. Efficient integration of fuzzy logic in data mining tools.
4. A hybridization of fuzzy sets with data mining techniques.

CONCLUSIONS

This chapter discussed how fuzzy logic can be used to solve several different data mining tasks, namely classification clustering, and discovery of association rules. The discussion focused mainly one representative algorithm for each of these tasks.

There are at least two motivations for using fuzzy logic in data mining, broadly speaking. First, as mentioned earlier, fuzzy logic can produce more abstract and flexible patterns, since many quantitative features are involved in data mining tasks. Second, the crisp usage of metrics is better replaced by fuzzy sets that can reflect, in a more natural manner, the degree of belongingness/membership to a class or a cluster.

REFERENCES

Cios, K. J., & Sztandera, L. M. (1992). Continuous ID3 algorithm with fuzzy entropy measures, Proc. IEEE lnternat. Con/i on Fuzz)' Systems, pp. 469-476.

Hong, T.P., Kuo, C.S. and Chi, S.C. (1999). A Fuzzy Data Mining Algorithm for Quantitative Values. Third International Conference on Knowledge-Based Intelligent Information Engineering Systems. Proceedings. IEEE, pp. 480-483.

Jang, J. (1994). Structure determination in fuzzy modeling: A fuzzy CART approach, in Proc. IEEE Conf. Fuzzy Systems, pp. 480–485.

Janikow, C.Z. (1998), Fuzzy Decision Trees: Issues and Methods, IEEE Transactions on Systems, Man, and Cybernetics, 28(1): 1-14.

Komem, J., & Schneider, M. (2005), On the Use of Fuzzy Logic in Data Mining, in The Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach (Eds.), Springer, pp. 517-533.

Maher, P. E., & Clair, D. C. (1993). Uncertain reasoning in an ID3 machine learning framework, in Proc. 2nd IEEE Int. Conf. Fuzzy Systems, pp. 7–12.

Mitra, S., & Hayashi, Y. (2000). Neuro-fuzzy Rule Generation: Survey in Soft Computing Framework. IEEE Trans. Neural Networks, 11(3):748-768.

Mitra, S., & Pal, S. K. (2005), Fuzzy sets in pattern recognition and machine intelligence, Fuzzy Sets and Systems 156(1):381–386

Nasraoui, O., & Krishnapuram, R. (1997). A Genetic Algorithm for Robust Clustering Based on a Fuzzy Least Median of Squares Criterion, Proceedings of NAFIPS, Syracuse NY, pp. 217-221.

Nauck, D. (1997). Neuro-Fuzzy Systems: Review and Prospects Paper appears in Proc. Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), Aachen, pp. 1044-1053

Olaru, C., & Wehenkel L. (2003). A complete fuzzy decision tree technique, Fuzzy Sets and Systems, 138(2):221-254, 2003.

Peng, Y. (2004). Intelligent condition monitoring using fuzzy inductive learning, Journal of Intelligent Manufacturing, 15 (3): 373-380.

Shnaider, E., Schneider, M., & Kandel A. (1997). A Fuzzy Measure for Similarity of Numerical Vectors, Fuzzy Economic Review, 2(1):17-38.

Yuan, Y., & Shaw M. (1995). Induction of fuzzy decision trees, Fuzzy Sets and Systems, 69(1):125-139.

Zimmermann H. J. (2005), Fuzzy Set Theory and its Applications, Springer, 4th edition.

KEY TERMS

Association Rules: Techniques that find in a database conjunctive implication rules of the form "X and Y implies A and B."

Attribute: A quantity describing an instance. An attribute has a domain defined by the attribute type, which denotes the values that can be taken by an attribute.

Classifier: A structured model that maps unlabeled instances to finite set of classes.

Clustering: The process of grouping data instances into subsets in such a manner that similar instances are grouped together into the same cluster, while different instances belong to different clusters.

Data Mining: The core of the KDD process, involving the inferring of algorithms that explore the data, develop the model, and discover previously unknown patterns.

Fuzzy Logic: A type of logic that recognizes more than simple true and false values. With fuzzy logic, propositions can be represented with degrees of truth-fulness and falsehood thus it can deal with imprecise or ambiguous data. Boolean logic is considered to be a special case of fuzzy logic.

Instance: A single object of the world from which a model will be learned, or on which a model will be used.

Knowledge Discovery in Databases (KDD): A nontrivial exploratory process of identifying valid, novel, useful, and understandable patterns from large and complex data repositories.

Independent Subspaces

Lei Xu
Chinese University of Hong Kong, Hong Kong, & Peking University, Beijing, China

INTRODUCTION

Several unsupervised learning topics have been extensively studied with wide applications for decades in the literatures of statistics, signal processing, and machine learning. The topics are mutually related and certain connections have been discussed partly, but still in need of a systematical overview. The article provides a unified perspective via a general framework of independent subspaces, with different topics featured by differences in choosing and combining three ingredients. Moreover, an overview is made via three streams of studies. One consists of those on the widely studied principal component analysis (PCA) and factor analysis (FA), featured by the second order independence. The second consists of studies on a higher order independence featured independent component analysis (ICA), binary FA, and nonGaussian FA. The third is called mixture based learning that combines individual jobs to fulfill a complicated task. Extensive literatures make it impossible to provide a complete review. Instead, we aim at sketching a roadmap for each stream with attentions on those topics missing in the existing surveys and textbooks, and limited to the authors' knowledge.

A GENERAL FRAMEWORK OF INDEPENDENT SUBSPACES

A number of unsupervised learning topics are featured by its handling on a fundamental task. As shown in Fig.1(b), every sample x is projected into $\hat{x}$ on a manifold and the error $e = x - \hat{x}$ of using $\hat{x}$ to represent x is minimized collectively on a set of samples. One widely studied situation is that a manifold is a subspace represented by linear coordinates, e.g., spanned by three linear independent basis vectors a_1, a_2, a_3 as shown in Fig.1(a). So, $\hat{x}$ can be represented by its projection $y^{(j)}$ on each basis vector, i.e.,

$$\hat{x} = \sum_j^3 y^{(1)} a_j$$

or

$$x = \hat{x} + e = Ay + e, \quad [y = \quad y^{(1)}, y^{(2)}, y^{(3)}]^T. \tag{1}$$

Typically, the error $e = x - \hat{x}$ is measured by the square norm, which is minimized when e is orthogonal to $\hat{x}$. Collectively, the minimization of the average error $\|e\|^2$ on a set of samples or its expectation $E\|e\|^2$ is featured by those natures given at the bottom of Fig.1(a).

Generally, the task consists of three ingredients, as shown in Fig.2. First, how the error $e = x - \hat{x}$ is measured. Different measures define different projections. The square norm $d = \|e\|^2$ applies to a homogeneous medium between x and $\hat{x}$. Other measures are needed for inhomogeneous mediums. In Fig.1(c), a non-orthogonal but still linear projection is considered via $d = \|e\|_B^2 = e^T \Sigma_e^{-1} e$ with $\Sigma_e^{-1} = B^T B$, as if e is first mapped to a homogeneous medium by a linear mapping e and then measured by the square norm. Shown at the bottom of Fig.1(c) are the natures of this $Min\|e\|_B^2$. Being considerably different from those of $Min\|e\|^2$, more assumptions have to be imposed externally.

The second ingredient is a coordinate system, via either linear vectors in Fig.1(a)&(c) or a set of curves on a nonlinear manifold in Fig.1(b). Moreover, there is the third ingredient that imposes certain structure to further constrict how y is distributed within the coordinates, e.g., by the nature d).

The differences in choosing and combining the three ingredients lead to different approaches. We use the name "independent subspaces" to denote those structures with the components of y being mutually independent, and get a general framework for accommodating several unsupervised learning topics.

Subsequently, we summarize them via three streams of studies by considering

- $d = \|e\|_B^2 = e^T \Sigma_e^{-1} e$ and two special cases,
- three types of independence structure, and whether there is temporal structure among samples,
- varying from one linear coordinate system to multiple linear coordinate systems at different locations, as shown in Fig.2.

Figure 1.

For $\min E\|e\|^2$, we have
Consequences
a) e from $G(e|0,\sigma_e^2 I)$, $\sigma_e^2 = \min E\|e\|^2$,
b) the coordinates in y is reachable by an orthogonal transform $y = Wx$
c) $Eey^T = 0$ (i.e., not correlated)
Assumptions
d) for a unique A, it needs to impose that a_1, a_2, a_3 are orthonormal $A^T A = I$, which implies the nature: $Eyy^T = \Lambda$ is diagonal, i.e., $y^{(1)}, y^{(2)}, y^{(3)}$ are of the 2nd order independence.
Indeterminacy
e) any rotation of the coordinate system leads to the same subspace, i.e., there is an indeterminacy of a rotation matrix Φ with $A' = \Phi A$: $A'^T A' = A^T \Phi^T \Phi A = A^T A = I$.

For $\min E\|e\|_B^2$, $\|e\|_B^2 = e^T \Sigma_e^{-1} e$, we have
Consequences
a) e from $G(e|0,\Sigma_e)$,
b) y is reached by a $y = Wx$, but W is non-orthogonal.
Assumptions
c) $Eey^T = 0$ (i.e., not correlated), here it is no longer a consequence.
d) $Eyy^T = \Lambda$ is diagonal, while $A^T A = I$ is removed because it impedes $E\|e\|_B^2$ to reach its minimum.
Indeterminacy
e) still a rotation matrix Φ since $A' = \Phi A$ spans the same subspace.
f) a diagonal D with $A' = AD$, $y' = D^{-1}y$: $Ay + e = A'y' + e$ & $Ey'y'^T$ is still diagonal.
g) a unknown allocation between the two additive terms in $Exx^T = A\Lambda A^T + \Sigma_e$.
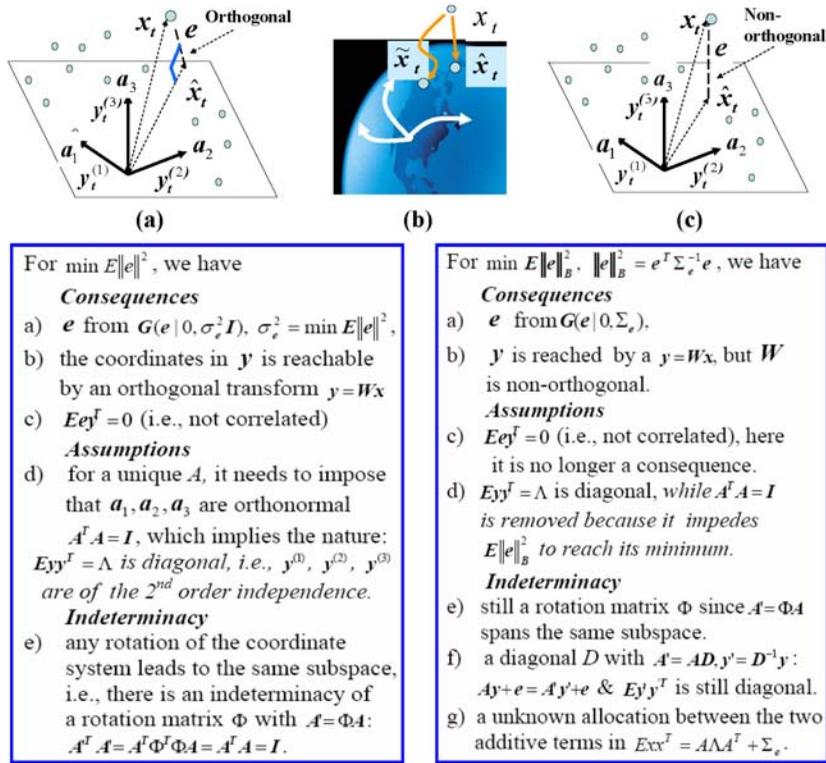
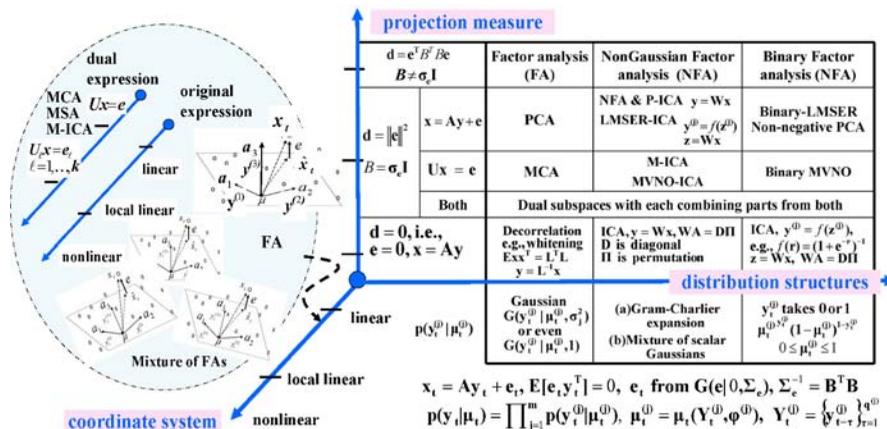**Fig.1 A General Framework of Independent Subspaces**

Figure 2.

Fig.2 Three gradients and their typical choices

STUDIES FEATURED BY SECOND ORDER INDEPENDENCE

We start at considering samples of independently and identically distributed (i.i.d.) by linear coordinates and an independent structure of a Gaussian $\mathbf{p}(\mathbf{y}_t^{(j)}|\mathbf{\lambda}^{(j)})$, with the projection measure varying as illustrated within the first column of the table in Fig.2. We encounter factor analysis (FA) in the general case $d = \|e\|_B^2 = e^T B^T Be$. At the special case $B = \sigma_e I$, the linear coordinates span a principal subspace of data. Further imposing $A^T A = I$ and requiring the columns of A given by the first m principal components (PCs), i.e., eigenvectors that correspond the largest eigenvalues of $\Sigma = (B^T B)^{-1}$. It becomes equivalent to PCA. Moreover, at the degenerated case $e = 0$, $y = xW$ de-correlates components of y, e.g., performing a pre-whitening as encountered in signal processing.

We summarize studies on the Roadmap A. The first stream originated from 100 years ago. The first adaptive learning one is Oja rule that finds the 1st-PC (i.e., the eigenvector that corresponds the largest eigenvalue of Σ), without explicitly estimating Σ. Extended to find multi-PCs, one way is featured by either an asymmetrical or a sequential implementation of the 1st-PC rule, but suffering error-accumulation. Details are referred to Refs.5,6,67,76,96 in (Xu, 2007a). The other way is finding multi-PCs symmetrically, e.g., Oja subspace rule. Further studies are summarized into the following branches:

MCA, Dual Subspace, and TLS Fitting

In (Xu, Krzyzak&Oja, 1991), a dual pattern recognition is suggested by considering both the principal subspace and its complementary subspace, as well as both the multiple PCs and its complementary counterparts--the components that correspond the smallest eigenvalues of Σ (i.e., the row vectors of U in Fig.2). Moreover, the first adaptive rule is proposed by eqn.(11a) in (Xu, Krzyzak&Oja, 1991) to get the component that corresponds the smallest eigenvalue of Σ, under the name Minor component analysis (MCA) firstly coined by Xu, Oja&Suen (1992), and it is also used for implementing a total least square (TLS) curve fitting. Subsequently, this topic has been brought to the signal processing literature by Gao, Ahmad & Swamy (1992) that was motivated by a visit of Gao to Xu's office where Xu introduced him the result of Xu,Oja&Suen (1992). Thereafter, adap-

tive MCA learning for TLS filtering becomes a popular topic of signal processing, see (Feng,Bao&Jiao,1998) and Refs.24,30,58,60 in (Xu,2007a).

It was also suggested in (Xu,Krzyzak&Oja,1992) that an implementation of PCA or MCA is made by switching the updating sign in the above eqn.(11a). Efforts were subsequently made to examine the existing PCA rules on whether they remain stable after such a sign switching. These jobs usually need tedious mathematical analyses of ODE stability, e.g., Chen & Amari (2001). An alternative way is turning an optimization of a PCA cost into a stable optimization of an induced cost for MCA, e.g., the LMSER cost is turned into one for subspace spanned by multiple MCs (Xu, 1994, see Ref.111, Xu2007a). A general method is further given by eqns(24-26) in (Xu, 2003) and then discussed in (Xu, 2007a).

LMSER Learning and Subspace Tracking

A new adaptive PCA rule is derived from the gradient $\nabla E_2(W)$ for a least mean square error reconstruction (LMSER) (Xu,1991), with the first proof proposed on global convergence of Oja subspace rule--a task that was previously regarded as difficult. It was shown mathematically and experimentally that LMSER improves Oja rule by further comparative studies, e.g, see (Karhunen,Pajunen&Oja,1998) and see (Refs14,15,48,54,71,72, Xu2007a). Two years after (Xu,1991), this $E_2(W)$ is used for signal subspace tracking via a recursive least square technique (Yang,1993), then followed by others in the signal processing literature (Refs.33&55, Xu2007a). Also, PCA and subspace analysis can be performed by other theories or costs (Xu, 1994a&b). The algebraic and geometric properties were further analyzed on one of them, namely relative uncertainty theory (RUT), by Fiori (2000&04, see Refs.25,29, Xu2007a). Moreover, the NIC criterion for subspace tracking is actually a special case of this RUT, which can be observed by comparing eqn.(20) in (Miao& Hua,1998) with the equation of ρ_e at the end of Sec.III.B in (Xu,1994a).
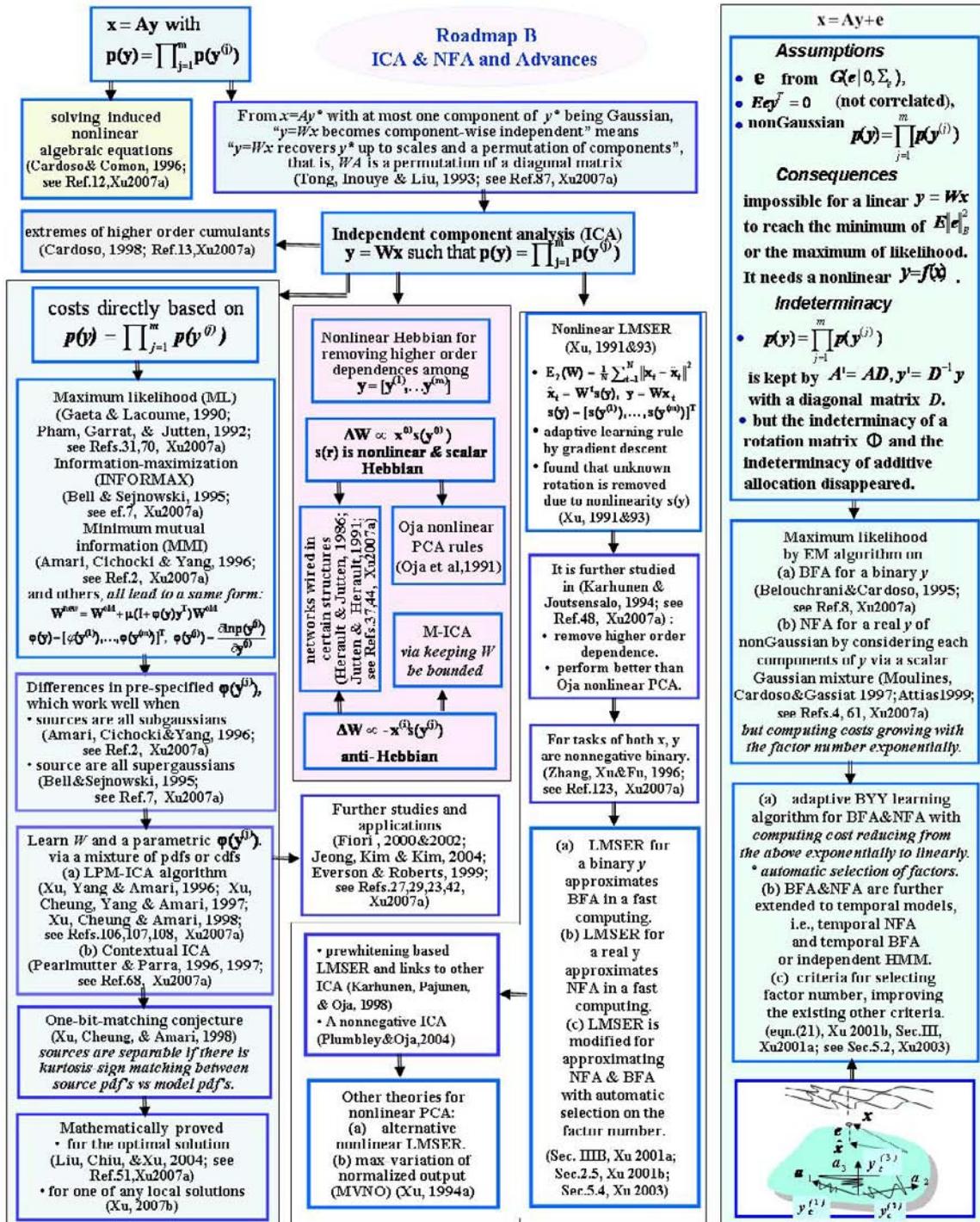
Principal Subspace vs. Multi-PCs

Oja subspace rule does not truly find the multi-PCs due to a rotation indeterminacy. Interestingly, it is demonstrated experimentally that adding a sigmoid function makes LMSER approximate the multi-PCs

Figure 3.
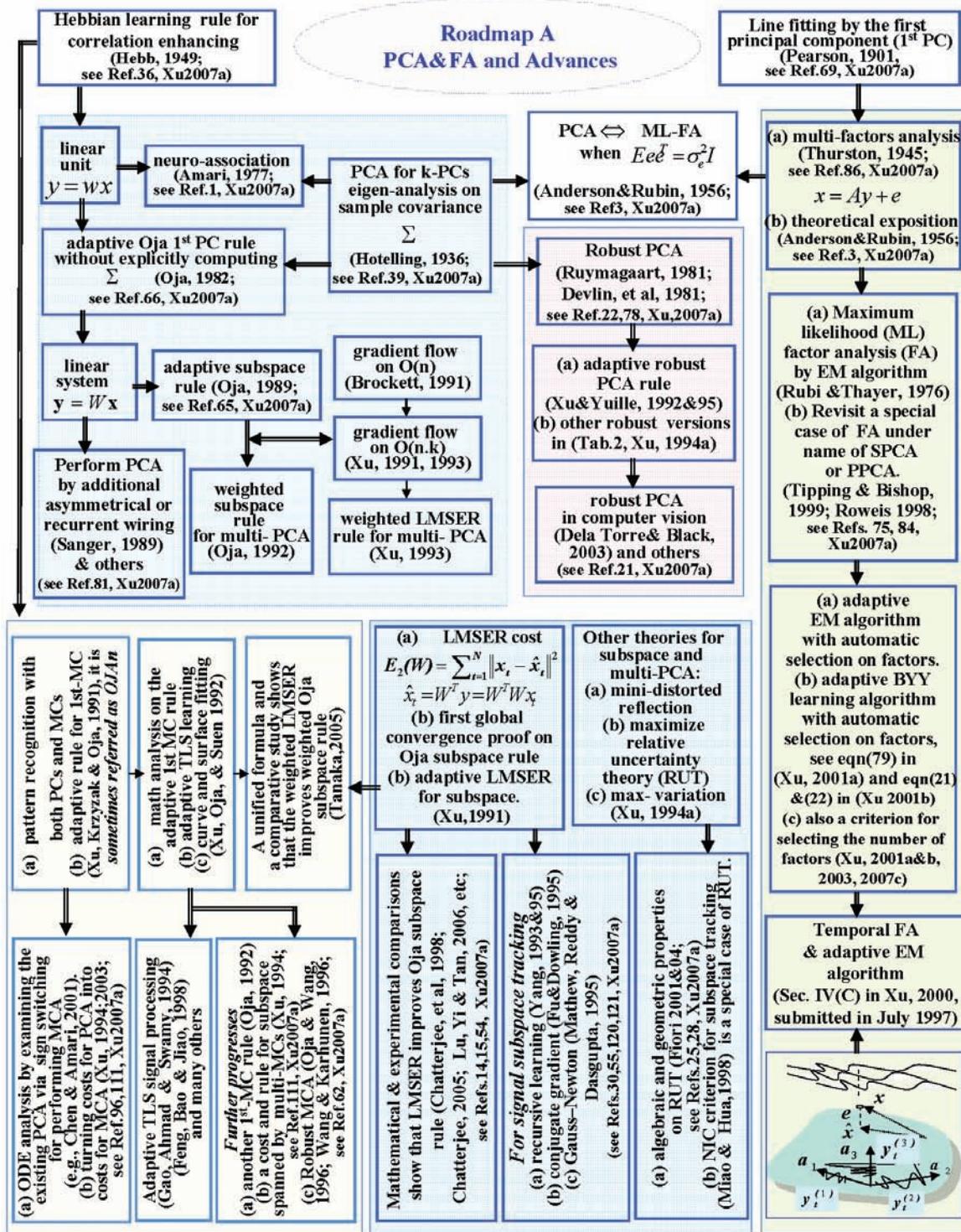
Figure 4.

well (Xu,1991). Working at Harvard in the late summer 1991, Xu got aware of Brockett (1991) and thus extended the Brockett flow of $n \times n$ orthogonal matrices to that of $n \times n_1$ orthogonal matrices with $n > n_1$, from which two learning rules for truly the multi-PCs are obtained through modifying the LMSER rule and Oja subspace rule. The two rules were included as eqns (13)&(14) in Xu (1993) that was submitted in 1991, which are independent and also different from Oja (1992). Recently, Tanaka (2005) unifies these rules into one expression controlled by one parameter, and a comparative study was made to show that eqn(14) in (Xu,1993) turned out to be the most promising one.

Adaptive Robust PCA

In the statistics literature, robust PCA was proposed to resist outliers via a robust estimator on Σ. Xu&Yuille (1992&95) generalized the rules of Oja, LMSER, and MCA into robust adaptive learning by statistical physics, related to the Huber M-estimators. Also, the PCA costs in (Xu,1994b) are extended to robust versions in Tab.2 of (Xu, 1994a). Thereafter, efforts have been further made, including its use in computer vision, e.g., see (Refs9,21,45,52, Xu2007a).

On Roadmap A, another branch consists of advances on FA, which includes PCA as its special case at $\Sigma_e = \sigma_e^2 I$. In the past decade, there is a renewed interest in FA, not only the EM algorithm for FA is brought to implementing PCA, but also adaptive EM algorithm and other advances are developed in help of the Bayesian Ying Yang (BYY) harmony learning.

SUBSPACES OF HIGHER ORDER INDEPENDENCE

Noticing the table in Fig.2, we proceed as $\mathbf{p}(\mathbf{y}_\mathbf{t}^{(\mathbf{i})} | \boldsymbol{\mu}^{(\mathbf{j})})$ becomes nonGaussian ones in the last two columns. Shown at the left-upper corner on Roadmap B, the degenerated case $e = 0$ leads to the problem of solving $x = Ay$ from samples of x and an independence constraint

$$p(y) = \prod_{j=1}^{m} p(y^{(j)})$$
.

One way is solving induced nonlinear algebraic equations. Another way is called independent component analysis (ICA), tackled in the following four branches:

- Seeking extremes of the higher order cumulants of y.
- Using nonlinear Hebbian learning for removing higher order dependences among components of y, actually from which ICA studies originate.
- Optimizing a cost that bases on

$$p(y) = \prod_{j=1}^{m} p(y^{(j)})$$

directly. As shown on Roadmap B, a same updating equation is reached from several aspects, with actual differences coming from pre-specifying the nonlinearity of $f(y^{(j)})$. One works when the source components of y^* are all subgaussians while the other works when the components of y^* are all supergaussians. This problem is solved by learning jointly W and $f(y^{(j)})$ via a parametric model. It is further found that a rough estimate of each source is already enough, which motivates the so called one-bit-matching conjecture that is recently proved to be true mathematically (Xu, 2007b).

- Implementing nonlinear LMSER (Xu, 1991&93). Details are referred to Roadmap B. Here, we add clarifications on two previous confusions. One relates to an omission of the origin of nonlinear LMSER. This has already been clarified in (Karhunen,Pajunen, &Oja,1998; Hyvarinen, Karhunen, & Oja, 2001;Plumbley &Oja,2004), clearly spelling out that the nonlinear $E_2(W)$ and its adaptive gradient rule were both proposed firstly in (Xu, 1991&93). The second confusion is about that ICA is usually regarded as a counterpart of PCA. As stated in (Xu,2001b&03) and observed from the Table in Fig.2, ICA by $y = xW$ is actually an extension of de-correlation analysis, in any combinations of PCs and MCs. The counterpart of MCA is minor ICA (M-ICA) while the counterpart of PCA is principal ICA (P-ICA).

In fact, the concept `principal' emerges from $e_t = x_t - Ay \neq 0$. As shown within the table in Fig.2 and on the rightmost column on Roadmap B, as $\mathbf{p}(\mathbf{y}_\mathbf{t}^{(\mathbf{i})} | \boldsymbol{\mu}^{(\mathbf{j})})$

becomes nonGaussian ones, FA is extended to a binary FA (BFA) if y is binary, and a nonGaussian FA (NFA) if y is real but nonGaussian. Similar to FA performing PCA at $\Sigma_e = \sigma_e^2 I$, both BFA and NFA become to perform a P-ICA at $\Sigma_e = \sigma_e^2 I$.

Observing the first box in this column, for $e_t = x_t - Ay \neq 0$ we need to seek an appropriate nonlinear map $y = f(x)$. It usually has no analytical solution but needs an expensive computation to approximate. As discussed in (Xu, 2003), nonlinear LMSER uses a sigmoid non-linearity $y_t^{(j)} = s(z_t^{(j)}), z = xW$ to avoid computing costs and approximately implements a BFA for a Bernoulli $p(y^{(j)})$ with a probability $p_j = \frac{1}{N}\sum_{t=1}^{N} s(z_t^{(j)})$ and a NFA for $p(y^{(j)})$ with a pseudo uniform distribution on $(-\infty, +\infty)$, as well as a nonnegative ICA (Plumbley&Oja,2004) when $p(y^{(j)})$ is on $[0, +\infty)$. However, further quantitative analysis is needed for this approximation.

Without approximation, the EM algorithm is developed for maximum likelihood learning since 1997, still suffering expensive computing costs. Favorably, further improvements have also been achieved by the BYY harmony learning. Details are referred to the rightmost column on Roadmap B.

TEMPORAL AND LOCALIZED EXTENSIONS

We further consider temporal samples shown at the bottom of the rightmost column on both Roadmap A and Roadmap B, via embedding a temporal structure in $\mathbf{p(y_t^{(j)} | \mu_t^{(j)})}$. A typical one is using

$$\mathbf{\mu_t^{(j)} = \mu^{(j)}(Y_t^{(j)}, \varphi_j)}, \quad \mathbf{Y_t^{(j)} = \left\{ y_{t-\tau}^{(j)} \right\}_{\tau=1,}^{q^{(j)}}}$$

e.g., a linear regression

$$\mathbf{\mu_t^{(j)} = \sum_{\tau=1}^{q^{(j)}} \beta_\tau^{(j)} y_{t-\tau}^{(j)},}$$

to turn a model (e.g., one in the table of Fig.2) into temporal extensions. Information is carried over time in two ways. One is computing $\mathbf{\mu_t^{(j)}}$ by the regression, with learning on $\mathbf{\mu_t^{(j)}}$ made through the gradient with respect to $\mathbf{J}_j$ by a chain rule. The second is computing $\int \mathbf{p(y_t^{(j)} | \mu_t^{(j)}) p(Y_t^{(j)}) dY_t^{(j)}}$ and getting the gradient with respect to $\mathbf{J}_j$. Details are referred to Xu (2000&01a&03).

Next, we move to multiple subspaces at different locations as shown in Fig.2. Studies are summarized on Roadmap C, categorized according to one key point, i.e., a scheme $p_{\ell,t}$ that allocates a sample x_t to different subspaces. This $p_{\ell,t}$ bases on two issues.

One is a local measure on how the ℓ-th subspace is suitable for representing x_t. The other is a mechanism that summarizes the local measures of subspaces to yield $p_{\ell,t}$. One typical mechanism is that emerges in the EM algorithm for the maximum likelihood or Bayesian learning, where x_t is fractionally allocated among subspaces proportional to their local measures. Another typical mechanism is that x_t is nonlinearly located to one or more winners via a competition based on the local measures, e.g., as in the classic competitive learning and the rival penalized competitive learning (RPCL).

Also, a scheme $p_{\ell,t}$ may come from blending both types of mechanisms, as that from the BYY harmony learning. Details are referred to (Xu,2007c) and its two http-sites.

FUTURE TRENDS

Another important task is how to determine the number k of subspaces and the dimension m_ℓ of each subspace. It is called model selection, usually implemented in two phases. First, a set of candidates are considered by enumerating k and m_ℓ, with unknown parameters estimated by the maximum likelihood learning. Second, the best among the candidates is selected by one of criteria, such as AIC, CAIC, SIC/BIC/MDL, Cross Validation, etc. However, this two-phase implementation is computationally very extensive. Moreover, the performance will degenerate considerably when the sample size is finite while k and m_ℓ are not too small.

One trend is letting model selection to be made automatically during learning, i.e., on a candidate with k and m_ℓ initially being large enough, learning not only determines unknown parameters but also automatically shrinks k and m_ℓ to appropriate ones. Two such efforts are RPCL and the BYY harmony learning. Details are referred to (Xu,2007c) and its two http-sites.

Also, there are open issues on $x = Ay + e$, $e \neq 0$, with components of y mutually independent in higher order statistics. Some are listed below:

Figure 5.

Roadmap C   Mixtures of Local Subspaces

Models Allocation		*p(y)* is Gaussian		*p(y)* is nonGaussian	
		PCA/FA/TFA	MCA/Surface fitting	ICA	NFA/BFA/LMSER and temporal extensions
Competitive winning and penalizing	Classic CL	Local PCA (Sec.3.2, Xu 1995) (Kambhatla & Leen, 1997; see Ref.46, Xu2007a) PCA competitive learning (Lopez-Rubio, et al 2004; see Ref.50, Xu2007a)	Local MCA by MML (Sec.4.1, Xu 1995; see Ref.110, Xu2007a)	Competitive ICA (eqn.37, Xu 2001b) Competitive Temporal ICA (eqn.88, Xu 2001a)	the degenerated cases from the ones below
	Rival Penalized CL & BYY	Local PCA, Local FA, Local LMSER (Sec.4.3, Xu 2001b) (eqns.14&16, Xu 1998; see Ref.103, Xu2007a) Local FA (Sec.3.3.2, Xu, 2007c) Local TFA (Xu, 2004; see Ref.95, Xu2007a)	Local MCA by MML (eqn.15, Xu, 1998, see Ref.103, Xu2007a) (Sec.4.2, Xu, 2001b)	Improved competitive ICA (Sec.4, Xu 2002; see Ref.98, Xu2007a)	Local BFA, NFA, LMSER (eqns.43&44, Xu 2001b) (Sec.4, Xu 2002; see Ref.98, Xu2007a) temporal extensions (Xu, 2004, see Ref.95, Xu2007a)
		with automatic selection on either or both of the number of subspaces and the dimensions of subspaces.			
Proportional weighting	Maximum Likelihood	Local PCA by a simplified EM (Sec.V(B)(D), Xu 1994b) Mixtures of FA (Ghahramani & Hinton, 1996; see Ref.35, Xu2007a) Mixtures of probabilistic PCA (Tipping & Bishop, 1999; see Ref.84, Xu2007a)	Local MCA by a simplified EM (Sec.V(C)(D), Xu 1994b) MCA Co-integration (Xu & Leung, 1998, see Ref.105, Xu2007a) Probabilistic MCA (Williams&Agakov,2002, see Ref.91, Xu2007a)	ICA mixture (Lee, Lewicki, & Sejnowski, 2000; see Ref.50, Xu2007a)	One possible way is getting extension from (Moulines, Cardoso & Gassiat 1997; Attias 1999; see Ref.4,61, Xu2007a) but with much expensive computing costs.
	Bayesian	Variational Mixture (Ghahramani & Beal, 2000; Utsugi & Kumagai, 2001; see Ref.34,90, Xu2007a)		Variational Mixture (Choudrey & Roberts 2003; see Ref.17, Xu2007a)	

- Which part of unknown parameters in $x = Ay + e$ can be determined uniquely ?
- Under which conditions, the independence

$$p(y) = \prod_{j=1}^{m} p(y^{(j)})$$

can be ensured in concept? Can it be further achieved by a learning algorithm?
- In what a sense, both ensuring

$$p(y) = \prod_{j=1}^{m} p(y^{(j)})$$

and the best reconstruction of x by $\hat{x} = Ay$ can be achieved simultaneously? If not, what is the best nonlinear $y = f(x)$ in term of both

$$p(y) = \prod_{j=1}^{m} p(y^{(j)})$$

and $e \neq 0$?
- Can such a best be obtained analytically or via an effective computing?

CONCLUSION

Studies of three closely related unsupervised learning streams have been overviewed in an extensive scope

and from a systematic perspective. A general framework of independent subspaces is presented, from which a number of learning topics are summarized via different features of choosing and combining the three basic ingredients.

ACKNOWLEDGMENT

REFERENCES

Brockett, R.W., (1991), Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems, Linear Algebra and Its Applications 146,79-91.

Chen, T., & Amari, S., (2001), Unified stabilization approach to principal and minor components extraction algorithms, Neural Networks 14(10),1377–1387.

Feng, D.Z., Bao, Z., & Jiao, L.C., (1998), Total least mean squares algorithm, IEEE Transactions Signal Processing 46,2122–2130.

Gao, K., Ahmad, M.O., & Swamy, M.N., (1992), Learning algorithm for total least-squares adaptive signal processing, Electronic Letters 28(4),430–432.

Hyvarinen, A., Karhunen, J., & Oja, E., (2001), Independent component analysis, John Wiley, NY, 2001.

Karhunen, J., Pajunen, P. & Oja , E., (1998), The nonlinear PCA criterion in blind source separation: relations with other approaches, Neurocomputing 22,5-20.

Miao, Y.F., & Hua, Y.B., (1998), Fast subspace tracking and neural network learning by a novel information criterion, IEEE Transactions Signal Processing 46,1967–79.

Oja, E., (1992), Principal components, minor components, and linear neural networks, Neural Networks 5,927-935.

Oja, E., Ogawa, H., & Wangviwattana, J., (1991), Learning in nonlinear constrained Hebbian networks, Proc.ICANN'91, 385-390.

Plumbley, M.D., & Oja, E., (2004), A "nonnegative PCA" algorithm for independent component analysis, IEEE Transactions Neural Networks 15(1),66-76.

Tanaka, T., (2005), Generalized weighted rules for principal components tracking, IEEE Transactions Signal Processing 53(4),1243- 1253.

Xu, L., (2007a), A unified perspective on advances of independent subspaces: basic, temporal, and local structures, Proc.6th.Intel.Conf.Machine Learning and Cybernetics, Hong Kong, 19-22 Aug.2007, 767-776.

Xu, L., (2007b), One-bit-matching ICA theorem, convex-concave programming, and distribution approximation for combinatorics, Neural Computation 19,546-569.

Xu, L., (2007c), A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving, Pattern Recognition 40,2129-2153. Also see http://www.scholarpedia.org/article/Rival_Penalized_Competitive_Learning http://www.scholarpedia.org/article/Bayesian_Ying_Yang_Learning.

Xu, L., (2003), Independent component analysis and extensions with noise and time: A Bayesian Ying-Yang learning perspective, Neural Information Processing Letters and Reviews 1(1),1-52.

Xu, L., (2001a), BYY harmony learning independent state space and generalized APT financial analyses, IEEE Transactions Neural Networks 12,822–849.

Xu, L., (2001b), An Overview on Unsupervised Learning from Data Mining Perspective, Advances in Self-Organizing Maps, Allison et al, Eds., Springer, 2001,181–210.

Xu, L., (2000), Temporal BYY learning for state space approach, hidden Markov model and blind source separation, IEEE Transactions Signal Processing 48,2132–2144.

Xu, L., Cheung, C.C., & Amari, S., (1998), Learned parametric mixture based ICA algorithm, Neurocomputing 22,69-80.

Xu, L., (1994a), Beyond PCA learning: from linear to nonlinear and from global representation to local representation, Proc.ICONIP94, Vol.2,943-949.

Xu, L., (1994b), Theories for unsupervised learning: PCA and its nonlinear extensions, Proc.IEEE ICNN94, Vol.II,1252-1257.

Xu, L., (1993), Least mean square error reconstruction principle for self-organizing neural-nets, Neural Networks 6,627–648.

Xu, L., Oja, E., & Suen, C.Y., (1992), Modified Hebbian learning for curve and surface fitting, Neural Networks 5,393-407.

Xu, L., & Yuille, A.L., (1992&95), Robust PCA learning rules based on statistical physics approach, Proc.IJCNN92-Baltimore, Vol.I:812-817. An extended version on IEEE Transactions Neural Networks 6,131–143.

Xu, L., (1991), Least MSE reconstruction for self-organization, Proc.IJCNN91-Singapore, Vol.3,2363-73.

Xu, L., Krzyzak, A., & Oja, E., (1991), A neural net for dual subspace pattern recognition methods, International Journal Neural Systems 2(3),169-184.

Yang, B., (1993), Subspace tracking based on the projection approach and the recursive least squares method, Proc.IEEE ICASSP93, Vol.IV,145–148.

KEY TERMS

BYY Harmony Learning: It is a statistical learning theory for a two pathway featured intelligent system via two complementary Bayesian representations of the joint distribution on the external observation and its inner representation, with both parameter learning and model selection determined by a principle that two Bayesian representations become best harmony. See http://www.scholarpedia.org/article/Bayesian_Ying_Yang_Learning.

Factor Analysis: A set of samples $\left\{\mathbf{x_t}\right\}_{t=1}^{N}$ is described by a linear model $x = Ay + \mu + e$, where μ is a constant, y and e are both from Gaussian and mutually uncorrelated, and components of y are called factors and mutually uncorrelated. Typically, the model is estimated by the maximum likelihood principle.

Independence Subspaces: It refers to a family of models, each of which consists of one or several subspaces. Each subspace is spanned by linear independent basis vectors and the corresponding coordinates are mutually independent.

Least Mean Square Error Reconstruction (LMSER): For an orthogonal projection x_t onto a subspace spanned by the column vectors of a matrix W, maximizing $\frac{1}{N}\sum_{t=1}^{N}(w^T\mathbf{x_t})^2$ subject to $w^t w = I$ is equivalent to minimizing the mean square error $\frac{1}{N}\sum_{t=1}^{N}\left\|x_t - \hat{x}_t\right\|^2$ by using the projection $\hat{\mathbf{x}}_\mathbf{t} = \mathbf{W}\mathbf{W}^\mathbf{T}\mathbf{x_t}$ as reconstruction of x_t, which is reached when W spans the same subspace spanned by the PCs.

Minor Component (MC): Being orthogonal complementary to the PC, the solution of $\mathbf{min}_{(w^t w=1)}\mathbf{J(w)} = \frac{1}{N}\sum_{t=1}^{N}(\mathbf{w^r x_t})^2 = \mathbf{w^T \Sigma\, w}$ is the MC, while the m-MCs are referred to the columns of W that minimizes $\mathbf{J}(W) = \frac{1}{N}\sum_{t=1}^{N}\|W^r\mathbf{x_t}\|^2 = Tr[W^\mathbf{T}\Sigma W]$ subject to $w^t w = I$.

Principal Component (PC): For samples $\left\{\mathbf{x_t}\right\}_{t=1}^{N}$ with a zero mean, its PC is a unit vector w originated at zero with a direction along which the average of the orthogonal projection by every sample is maximized, i.e., $\mathbf{max}_{(w^t w=1)}\mathbf{J(w)} = \frac{1}{N}\sum_{t=1}^{N}(\mathbf{w}^T\mathbf{x_t})^2 = \mathbf{w^T\Sigma\, w}$, the solution is the eigenvector of the sample covariance matrix $\mathbf{\Sigma} = \frac{1}{N}\sum_{t=1}^{N}\mathbf{x_t x_t^T}$, corresponding to the largest eigen-value. Generally, the m-PCs are referred to the m orthonormal vectors as the columns of W that maximizes $\mathbf{J}(W) = \frac{1}{N}\sum_{t=1}^{N}\|W^r\mathbf{x_t}\|^2 = Tr[W^\mathbf{T}\Sigma W]$.

Rival Penalized Competitive Learning: It is a development of competitive learning in help of an appropriate balance between participating and leaving mechanisms, such that an appropriate number of agents or learners will be allocated to learn multiple structures underlying observations. See http://www.scholarpedia.org/article/Rival_Penalized_Competitive_Learning.

Total Least Square (TLS) Fitting: Given samples $\left\{\mathbf{z_t}\right\}_{t=1}^{N}$, $\mathbf{z_t} = [\mathbf{y_t}, \mathbf{x_t^T}]^\mathbf{T}$, instead of finding a vector w to minimize the error $\frac{1}{N}\sum_{t=1}^{N}\left\|y_t - w^T x_t\right\|^2$, the TLS fitting is finding an augmented vector $\tilde{w} = [w^T, c]^T$ such that the error $\frac{1}{N}\sum_{t=1}^{N}\left\|\tilde{w}^T z_t\right\|^2$ is minimized subject to $\tilde{w}^T\tilde{w} = 1$, the solution is the MC of $\left\{\mathbf{z_t}\right\}_{t=1}^{N}$.

I

Information Theoretic Learning

Deniz Erdogmus
Northeastern University, USA

Jose C. Principe
University of Florida, USA

INTRODUCTION

Learning systems depend on three interrelated components: topologies, cost/performance functions, and learning algorithms. Topologies provide the constraints for the mapping, and the learning algorithms offer the means to find an optimal solution; but the solution is optimal with respect to what? Optimality is characterized by the criterion and in neural network literature, this is the least addressed component, yet it has a decisive influence in generalization performance. Certainly, the assumptions behind the selection of a criterion should be better understood and investigated.

Traditionally, least squares has been the benchmark criterion for regression problems; considering classification as a regression problem towards estimating class posterior probabilities, least squares has been employed to train neural network and other classifier topologies to approximate correct labels. The main motivation to utilize least squares in regression simply comes from the intellectual comfort this criterion provides due to its success in traditional linear least squares regression applications – which can be reduced to solving a system of linear equations. For nonlinear regression, the assumption of Gaussianity for the measurement error combined with the maximum likelihood principle could be emphasized to promote this criterion. In nonparametric regression, least squares principle leads to the conditional expectation solution, which is intuitively appealing. Although these are good reasons to use the mean squared error as the cost, it is inherently linked to the assumptions and habits stated above. Consequently, there is information in the error signal that is not captured during the training of nonlinear adaptive systems under non-Gaussian distribution conditions when one insists on second-order statistical criteria. This argument extends to other linear-second-order techniques such as principal component analysis (PCA), linear discriminant analysis (LDA), and canonical correlation analysis (CCA). Recent work tries to generalize these techniques to nonlinear scenarios by utilizing kernel techniques or other heuristics. This begs the question: *what other alternative cost functions could be used to train adaptive systems and how could we establish rigorous techniques for extending useful concepts from linear and second-order statistical techniques to nonlinear and higher-order statistical learning methodologies*?

BACKGROUND

This seemingly simple question is at the core of recent research on information theoretic learning (ITL) conducted by the authors, as well as research by others on alternative optimality criteria for robustness to outliers and faster convergence, such as different L_p-norm induced error measures (Sayed, 2005), the epsilon-insensitive error measure (Scholkopf & Smola, 2001), Huber's robust m-estimation theory (Huber, 1981), or Bregman's divergence based modifications (Bregman, 1967). Entropy is an uncertainty measure that generalizes the role of variance in Gaussian distributions by including information about the higher-order statistics of the probability density function (pdf) (Shannon & Weaver, 1964; Fano, 1961; Renyi, 1970; Csiszár & Körner, 1981). For on-line learning, information theoretic quantities must be estimated nonparametrically from data. A nonparametric expression that is differentiable and easy to approximate stochastically will enable importing useful concepts such as stochastic gradient learning and backpropagation of errors. The natural choice is kernel density estimation (KDE) (Parzen, 1967), due its smoothness and asymptotic properties. The plug-in estimation methodology (Gyorfi & van der Meulen, 1990) combined with definitions of Renyi (Renyi, 1970), provides a set of tools that are well-tuned for learning applications – tools suitable

for supervised and unsupervised, off-line and on-line learning. Renyi's definition of entropy for a random variable *X* is

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int p^\alpha(x)dx \quad (1)$$

This generalizes Shannon's linear additivity postulate to exponential additivity resulting in a parametric family. Dropping the logarithm for optimization simplifies algorithms. Specifically of interest is the quadratic entropy (α=2), because its sample estimator requires only one approximation (the density estimator itself) and an analytical expression for the integral can be obtained for kernel density estimates. Consequently, a sample estimator for quadratic entropy can be derived for Gaussian kernels of standard deviation σ on an iid sample set $\{x_1,\ldots,x_N\}$ as the sum of pairwise sample (particle) interactions (Principe et al, 2000):

$$\hat{H}_2(X) = -\log(\frac{1}{N^2} \sum_{i=i}^{N} \sum_{j=1}^{N} G_{\sqrt{2}\sigma}(x_i - x_j)) \quad (2)$$

The pairwise interaction of samples through the kernel intriguingly provides a connection to entropy of particles in physics. Particles interacting trough *information forces* (as in the *N*-body problem in physics) can employ computational techniques developed for simulating such large scale systems. The use of entropy in training multilayer structures can be studied in the backpropagation of information forces framework (Erdogmus et al, 2002). The quadratic entropy estimator was employed in measuring divergences between probability densities and blind source separation (Hild et al, 2006), blind deconvolution (Lazaro et al, 2005), and clustering (Jenssen et al, 2006). Quadratic expressions with mutual-information-like properties were introduced based on the Euclidean and Cauchy-Schwartz distances (ED/CSD). These are advantageous with computational simplicity and statistical stability in optimization (Principe et al, 2000).

Following the conception of information potential and force and principles, the pairwise-interaction estimator is generalized to use arbitrary kernels and any order α of entropy. The stochastic information gradient (SIG) is developed (Erdogmus et al, 2003) to train adaptive systems with a complexity comparable to the LMS (least-mean-square) algorithm - essential for training complex systems with large data sets. Supervised and unsupervised learning is unified under information-based criteria. Minimizing error entropy in supervised regression or maximizing output entropy for unsupervised learning (factor analysis), minimization of mutual information between the outputs of a system to achieve independent components or maximizing mutual information between the outputs and the desired responses to achieve optimal subspace projections in classification is possible. Systematic comparisons of ITL with conventional MSE in system identification verified the advantage of the technique for nonlinear system identification and blind equalization of communication channels. Relationships with instrumental variables techniques were discovered and led to the error-whitening criterion for unbiased linear system identification in noisy-input-output data conditions (Rao et al, 2005).

SOME IDEAS IN AND APPLICATIONS OF ITL

Kernel Machines and Spectral Clustering: KDE has been motivated by the smoothness properties inherent to reproducing kernel Hilbert spaces (RKHS). Therefore, a practical connection between KDE-based ITL, kernel machines, and spectral machine learning techniques was imminent. This connection was realized and exploited in recent work that demonstrates an information theoretic framework for pairwise similarity (spectral) clustering, especially normalized cut techniques (Shi & Malik, 2000). Normalized cut clustering is shown to determine an *optimal* solution that maximizes the CSD between clusters (Jenssen, 2004). This connection immediately allows one to approach kernel machines from a density estimation perspective, thus providing a robust method to select the *kernel size*, a problem still investigated by some researchers in the kernel and spectral techniques literature. In our experience, kernel size selection based on suitable criteria aimed at obtaining the *best* fit to the training data - using Silverman's regularized squared error fit (Silverman, 1986) or leave-one-out cross-validation maximum likelihood (Duin, 1976), for instance - has proved to be convenient, robust, and accurate techniques that avoid many of the computational complexity and load

issues. Local data spread based modifications resulting in variable-width KDE are also observed to be more robust to noise and outliers.

An illustration of ITL clustering by maximizing the CSD between the two estimated clusters is provided in Figure 1. The samples are labeled to maximize

$$D_{CS}(p,q) = -\log \frac{< p,q >_f}{\| p \|_f \| q \|_f} \tag{3}$$

where p and q are KDE for two candidate clusters, f is the overall data KDE and the weighted inner product to measure angular distance between clusters is

$$< p,q >_f = \int p(x)q(x)f^{-1}(x)dx \tag{4}$$

When estimated using a weighted KDE variant, this criterion becomes equivalently

$$D_{CS}(p,q) \approx \frac{\sum\limits_{x_i \in p, y_j \in q} K_{1/f}(x_i, y_j)}{\sqrt{\sum\limits_{x_i \in p, x_i \in p} K_{1/f}(x_i, x_j) \sum\limits_{y_i \in q, y_j \in q} K_{1/f}(y_i, y_j)}} \tag{5}$$

where $K_{1/f}$ is an equivalent kernel generated from the original kernel K (Gaussian here). One difficulty with kernel machines is their nonparametric nature, the requirement to solve for the eigendecomposition of a large positive-definite matrix that has size $N \times N$, for N training samples. The solution is a weighted sum of kernels evaluated over each training sample, thus the test procedure for each novel sample involves evaluating the sum of N kernels: $y_{test} = \sum_{k=1}^{N} w_k K(x_{test} - x_k)$. The Fast Gauss Transform (FGT) (Greengard, 1991), which uses the polynomial expansions for a Gaussian (or other) kernel has been employed to overcome this difficulty. FGT carefully selects few center points around which truncated Hermite polynomial expansions approximate the kernel machine. FGT still requires heavy computational load in off-line training (minimum $O(N^2)$, typically $O(N^3)$). The selection of expansion centers is typically done via clustering (e.g., Ozertem & Erdogmus, 2006).

Correntopy as a Generalized Similarity Metric: The main feature of ITL is that it preserves the universe of concepts we have in neural computing, but allows the adaptive system to *extract more information* from the data. For instance, the general Hebbian principle is

Figure 1. Maximum CSD clustering of two synthetic benchmarks: training and novel test data (left), KDE using Gaussian kernels with Silverman-kernel-size (center), and spectral projections of data on two dominant eigenfunctions of the kernel. The eigenfunctions are approximated using the Nystrom formula.
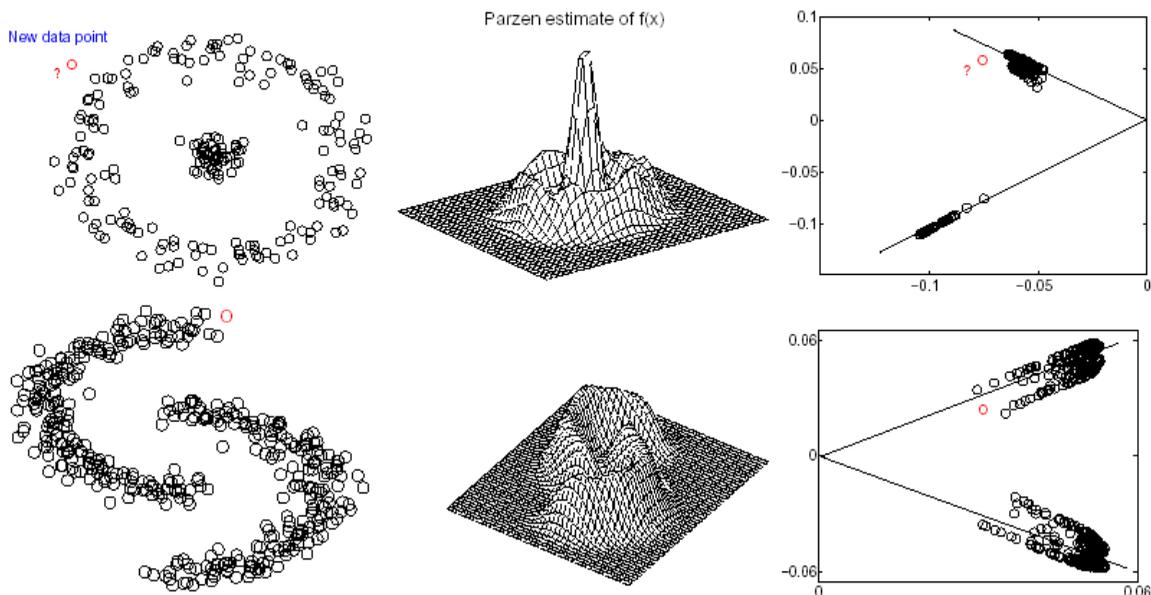
reduced into a second order metric in traditional artificial neural network literature (input-output product), thus becoming a synonym for second order statistics. The learning rule that maximizes output entropy (instead of output variance), using SIG with Gaussian kernels is $\Delta w(n) = \eta (x(n) - x(n-1))(y(n) - y(n-1))$ (Erdogmus et al, 2002), which still obeys the Hebbian principle, yet extracts more information from the data (leading to the error-whitening criterion for input-noise robust learning).

ITL quantifies global properties of the data, but will it be possible to apply it to functions, specifically those in RKHS? A concrete example is on similarity between random variables, which is typically expressed as second order correlation. Correntropy generalizes similarity to include higher order moment information. The name indicates the strong relation to correlation, but also stresses the difference – the average over the lags (for random processes) or over dimensions (for multidimensional random variables) is the information potential, i.e. the argument of second order Renyi's entropy. For random variables X and Y with joint density $p(x,y)$, correntropy is defined as

$$V(X,Y) = \int \int \delta (x-y) p(x,y) dx dy \tag{6}$$

and measures how dense the two random variables are along the line $x=y$ in the joint space. Notice that it is similar to correlation, which also asks the same question in a second moment framework. However, correntropy is local to the line $x=y$, while correlation is quadratically dependent upon distances of samples in the joint space. Using a KDE with Gaussian kernels

$$V(X,Y) = \frac{1}{N} \sum_{i=1}^{N} G(x_i - y_i) \tag{7}$$

Correntropy is a positive-definite function, thus defines a RKHS. Unlike correlation, RKHS is nonlinearly related to the input, because all moments of the random variable are included in the transformation. It is possible to analytically solve for least squares regression and principal components in this space, yielding nonlinear fits in input space. Correntopy induced metric (CIM) behaves as the L_2-norm for small distances and progressively approaches the L_1-norm and then converges to L_0 at infinity. Thus robustness to outliers is automatically achieved and equivalence to Huber's robust estimation can be proven (Santamaria, 2006). Unlike conventional kernel methods, correntropy solutions remain in the same dimensionality as the in-

Figure 2. Maximum mutual information projection versus kernel LDA test ROC results on hand-written digit recognition shown in terms of type-1 and type-2 errors (left); ROC results (P_{detect} vs P_{false}) compared for various techniques on sonar data. Both data are from the UCI Machine Learning Repository (2007).
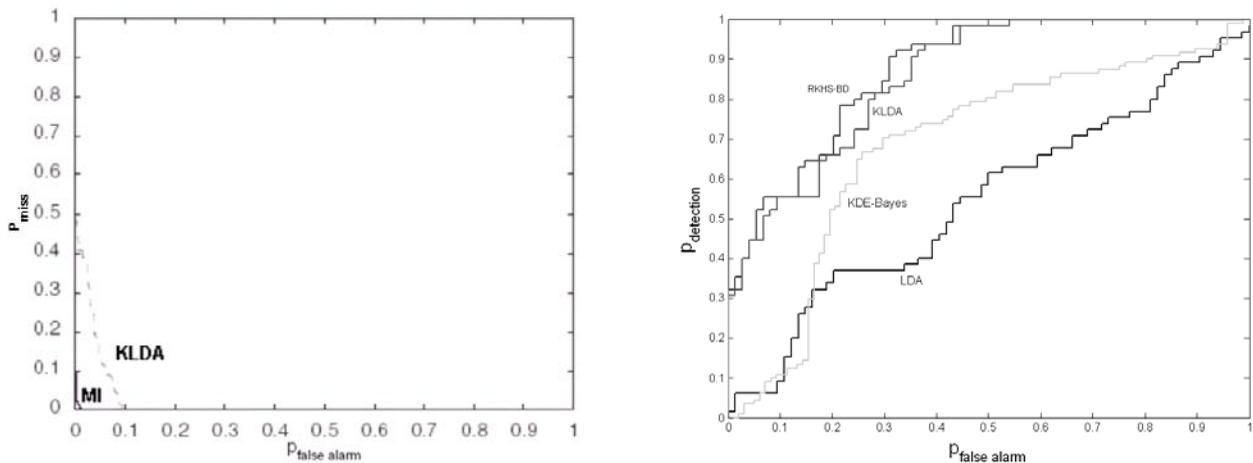
put vector. This might indicate built-in regularization properties, yet to be explored.

Nonparametric Learning in the RKHS: It is possible to obtain robust solutions to a variety of problems in learning using the nonparametric and local nature of KDE and its relationship with RKHS theory. Recently, we explored the possibility of designing nonparametric solutions to the problem of identifying nonlinear dimensionality reduction schemes that maintain maximal discriminative information in a pattern recognition problem (quite appropriately measured by the mutual information between the data and the class labels as agreed upon by many researchers). Using the RKHS formalism and based on the KDE, results were obtained that consistently outperformed the alternative rather heuristic kernel approaches such as kernel PCA and kernel LDA (Scholkopf & Smola, 2001). The conceptual oversight in the latter two is that, both PCA and LDA procedures are most appropriate for Gaussian distributed data (although acceptable for other symmetric unimodal distributions and are commonly but possibly inappropriately used for arbitrary data distributions).

Clearly, the distribution of the data in the kernel induced feature space could not be Gaussian for all typically exploited kernel selections (such as the Gaussian kernel), since these are usually translation invariant, therefore the data is, in principle, mapped to an infinite dimensional hypersphere on which the data could not have been Gaussian distributed (nor symmetrically distributed in general for the ideal kernel for a given problem since these are positive definite functions). Consequently, the hasty use of kernel extensions of second-order techniques is not necessarily optimal in a meaningful statistical sense. Nevertheless, these techniques have found *successful* applications in various problems; however, their suboptimality is clear from comparisons with more carefully designed solutions. In order to illustrate how drastic the performance difference could be, we present a comparison of a mutual information based nonlinear nonparametric projection approach (Ozertem et al, 2006) and kernel LDA in a simplified two-class handwritten digit classification case study and sonar mine detection case study. The ROC curves of both algorithms on the test set after being trained with the same data is shown in Figure 2. The kernel is assumed to be a circular Gaussian with size set to Silverman's rule-of-thumb. For the sonar data, we also include KDE-based approximate Bayes classifier and linear LDA for reference. In this example, KLDA performs close to mutual information projections, as observed occasionally.

FUTURE TRENDS

Nonparametric Snakes, Principal Curves and Surfaces: More recently, we have been investigating

Figure 3. Nonparametric snake after convergence from an initial state that was located at the boundary of the guitar image rectangle (left). The global principal curve of a mixture of ten Gaussians obtained according to the local subspace maximum definition for principal manifolds (right).
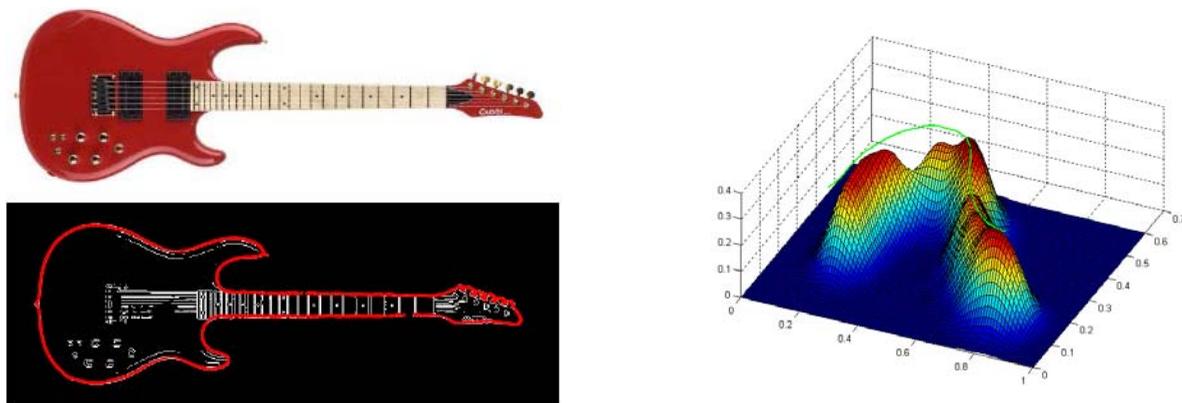
the application of KDE and RKHS to nonparametric clustering, principal curves and surfaces. Interesting mean-shift-like fixed-point algorithms have been obtained; specifically interesting is the concepts of *nonparametric snakes* (Ozertem & Erdogmus, 2007) and *local principal manifolds* (Erdogmus & Ozertem, 2007) that we developed recently. The nonparametric snake approach overcomes the principal difficulties experienced by snakes (active contours) for image segmentation, such as low capture range, data curvature inhomogeneity, and noisy and missing edge information. Similarly, the local conditions for determining whether a point is in a principal manifold or not provide guidelines for designing fixed point and other iterative learning algorithms for identifying such important structures.

Specifically in nonparametric snakes, we treat the edgemap of an image as samples and the values of the edginess as weights to construct a weighted KDE, from which, a fixed point iterative algorithm can be devised to detect the boundaries of an object in background. The designed algorithm can be easily made robust to outlier edges, converges very fast, and can penetrate into concavities, while not being trapped into the object at missing edge localities. The guitar image in Figure 3 emphasizes these advantages as the image exhibits both missing edges and concavities, while background complexity is trivially low as that was not the main concern in this experiment – the variable width KDE easily avoids textured obstacles. The algorithm could be utilized to detect the ridge-boundary of a structure in any dimensional data set in other applications.

In defining principal manifolds, we avoided the traditional least-squares error reconstruction type criteria, such as Hastie's self-consistent principal curves (Hastie & Stuetzle, 1989), and proposed a local subspace maximum definition for principal manifolds inspired by differential geometry. This definition lends itself to a uniquely defined principal manifold hierarchy such that one can use inflation and deflation to obtain a d-dimensional principal manifold from a $(d+1)$-dimensional principal manifold. The rigorous and local definition lends itself to easy algorithm design and multiscale principal structure analysis for probability densities. We believe that in the near future, the community will be able to prove maximal information preserving properties of principal manifolds obtained using this definition in a manner similar to mean-shift clustering solving for minimum information distortion clustering (Rao et al, 2006) and maximum likelihood modelling achieving minimum Kullback-Leibler divergence asymptotically (Carreira-Perpinan & Williams, 2003; Erdogmus & Principe, 2006).

CONCLUSION

The use of information theoretic learning criteria in neural networks and other adaptive system solutions have so far clearly demonstrated a number of advantages that arise due to the increased information content of these measures relative to second-order statistics (Erdogmus & Principe, 2006). Furthermore, the use of kernel density estimation with smooth kernels allows one to obtain continuous and differentiable criteria suitable for iterative descent/ascent-based learning and the nonparametric nature of KDE and its variants (such as variable-size kernels) allow one to achieve simultaneously robustness, global optimization through *kernel annealing*, and data modeling flexibility in designing neural networks and learning algorithms for a variety of benchmark problems. Due to lack of space, detailed mathematical treatments cannot be provided in this article; the reader is referred to the literature for details.

REFERENCES

Bregman, L.M., (1967). The Relaxation Method of Finding the Common Point of Convex Sets and Its Application to the Solution of Problems in Convex Programming. *USSR Computational Mathematics and Physics*, (7), 200-217.

Carreira-Perpinan, M.A., Williams, C.K.I., (2003). On the Number of Modes of a Gaussian Mixture. *Proceedings of Scale-Space Methods in Computer Vision*. 625-640.

Csiszár, I., Körner, J. (1981). *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Academic Press.

Duin, R.P.W., On the Choice of Smoothing Parameter for Parzen Estimators of Probability Density Functions. *IEEE Transactions on Computers*, (25) 1175-1179.

Erdogmus, D., Ozertem, U., (2007). Self-Consistent Locally Defined Principal Surfaces. *Proceedings of ICASSP 2007*. to appear.

Erdogmus, D., Principe, J.C., From Linear Adaptive Filtering to Nonlinear Information Processing. *IEEE Signal Processing Magazine*, (23) 6, 14-33.

Erdogmus, D., Principe, J.C., Hild II, K.E., (2002). Do Hebbian Synapses Estimate Entropy? *Proceedings of NNSP'02*, 199-208.

Erdogmus, D., Principe, J.C., Hild II, K.E., (2003). On-Line Entropy Manipulation: Stochastic Information Gradient. *IEEE Signal Processing Letters*, (10) 8, 242-245.

Erdogmus, D., Principe, J.C., Vielva, L. Luengo, D., (2002). Potential Energy and Particle Interaction Approach for Learning in Adaptive Systems. *Proceedings of ICANN'02*, 456-461.

Fano, R.M. (1961). *Transmission of Information: A Statistical Theory of Communications*, MIT Press.

Greengard, L., Strain, J., (1991). The Fast Gauss Transform. *SIAM Journal of Scientific and Statistical Computation*, (12) 1, 79–94.

Gyorfi, L., van der Meulen, E.C. (1990). On Nonparametric Estimation of Entropy Functionals. *Nonparametric Functional Estimation and Related Topics*, (G. Roussas, ed.), Kluwer Academic Publisher, 81-95.

Hastie, T., Stuetzle, W., (1989). Principal Curves. *Journal of the American Statistical Association*, (84) 406, 502-516.

Hild II, K.E., Erdogmus, D., Principe, J.C., (2006). An Analysis of Entropy Estimators for Blind Source Separation. *Signal Processing*, (86) 1, 182-194.

Huber, P.J., (1981). *Robust Statistics*. Wiley.

Jenssen, R., Erdogmus, D., Principe, J.C., Eltoft, T., (2004). The Laplacian PDF Distance: A Cost Function for Clustering in a Kernel Feature Space. *Advances in NIPS'04*, 625-632.

Jenssen, R., Erdogmus, D., Principe, J.C., Eltoft, T., (2006). Some Equivalences Between Kernel Methods and Information Theoretic Methods. *JVLSI Signal Processing Systems*, (45) 1-2, 49-65.

Lazaro, M., Santamaria, I., Erdogmus, D., Hild II, K.E., Pantaleon, C., Principe, J.C., (2005). Stochastic Blind Equalization Based on PDF Fitting Using Parzen Estimator. *IEEE Transactions on Signal Processing*, (53) 2, 696-704.

Ozertem, U., Erdogmus, D., (2006). Maximum Entropy Approximation for Kernel Machines. *Proceedings of MLSP 2005*.

Ozertem, U., Erdogmus, D., Jenssen, R., (2006). Spectral Feature Projections that Maximize Shannon Mutual Information with Class Labels. *Pattern Recognition*, (39) 7, 1241-1252.

Ozertem, U., Erdogmus, D., (2007). A Nonparametric Approach for Active Contours. *Proceedings of IJCNN 2007*, to appear.

Parzen, E., (1967). On Estimation of a Probability Density Function and Mode. *Time Series Analysis Papers*, Holden-Day, Inc.

Principe, J.C., Fisher, J.W., Xu, D., (2000). Information Theoretic Learning. *Unsupervised Adaptive Filtering*, (S. Haykin, ed.), Wiley, 265-319.

Rao, Y.N., Erdogmus, D., Principe, J.C., (2005). Error Whitening Criterion for Adaptive Filtering: Theory and Algorithms. *IEEE Transactions on Signal Processing*, (53) 3, 1057-1069.

Rao, S., de Madeiros Martins, A., Liu, W., Principe, J.C., (2006). Information Theoretic Mean Shift Algorithm. *Proceedings of MLSP 2006*.

Renyi, A., (1970). *Probability Theory*, North-Holland Publishing Company.

Sayed, A.H. (2005). *Fundamentals of Adaptive Filtering*. Wiley & IEEE Press.

Scholkopf, B., Smola, A.J. (2001). *Learning with Kernels*. MIT Press.

Shannon, C.E., Weaver, W. (1964). *The Mathematical Theory of Communication*, University of Illinois Press.

Shi, J., Malik, J., (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (22) 8, 888-905.

Silverman, B.W., (1986). *Density Estimation for Statistics and Data Analysis,* Chapman & Hall.

Santamaria, I., Pokharel, P.P., Principe, J.C., (2006). Generalized Correlation Function: Definition, Properties, and Application to Blind Equalization. *IEEE Transactions on Signal Processing***, (54) 6, 2187-2197.**

UCI Machine Learning Repository (2007). *http://mlearn.ics.uci.edu/MLRepository.html.* last accessed in June 2007.

KEY TERMS

Cauchy-Schwartz Distance: An angular density distance measure in the Euclidean space of probability density functions that approximates information theoretic divergences for nearby densities.

Correntropy: A statistical measure that estimates the similarity between two or more random variables by integrating the joint probability density function along the main diagonal of the vector space (line along ones). It relates to Renyi's entropy when averaged over sample-index lags.

Information Theoretic Learning: A technique that employs information theoretic optimality criteria such as entropy, divergence, and mutual information for learning and adaptation.

Information Potentials and Forces: Physically intuitive pairwise particle interaction rules that emerge from information theoretic learning criteria and govern the learning process, including backpropagation in multilayer system adaptation.

Kernel Density Estimate: A nonparametric technique for probability density function estimation.

Mutual Information Projections: Maximally discriminative nonlinear nonparametric projections for feature dimensionality reduction based on the reproducing kernel Hilbert space theory.

Renyi Entropy: A generalized definition of entropy that stems from modifying the additivity postulate and results in a class of information theoretic measures that contain Shannon's definitions as special cases.

Stochastic Information Gradient: Stochastic gradient of nonparametric entropy estimate based on kernel density estimation.

Intelligent Classifier for Atrial Fibrillation (ECG)

O. Valenzuela
University of Granada, Spain

I. Rojas
University of Granada, Spain

F. Rojas
University of Granada, Spain

A. Guillen
University of Granada, Spain

L. J. Herrera
University of Granada, Spain

F. J. Rojas
University of Granada, Spain

M. Cepero
University of Granada, Spain

INTRODUCTION

This chapter is focused on the analysis and classification of arrhythmias. An arrhythmia is any cardiac pace that is not the typical sinusoidal one due to alterations in the formation and/or transportation of the impulses. In pathological conditions, the depolarization process can be initiated outside the sinoatrial (SA) node and several kinds of extra-systolic or ectopic beatings can appear.

Besides, electrical impulses can be blocked, accelerated, deviated by alternate trajectories and can change its origin from one heart beat to the other, thus originating several types of blockings and anomalous connections. In both situations, changes in the signal morphology or in the duration of its waves and intervals can be produced on the ECG, as well as a lack of one of the waves.

This work is focused on the development of intelligent classifiers in the area of biomedicine, focusing on the problem of diagnosing cardiac diseases based on the electrocardiogram (ECG), or more precisely on the differentiation of the types of atrial fibrillations. First of all we will study the ECG, and the treatment of the ECG in order to work with it, with this specific pathology. In order to achieve this we will study different ways of elimination, in the best possible way, of any activity that is not caused by the auriculars. We will study and imitate the ECG treatment methodologies and the characteristics extracted from the electrocardiograms that were used by the researchers that obtained the best results in the Physionet Challenge, where the classification of ECG recordings according to the type of Atrial Fibrillation (AF) that they showed, was realised. We will extract a great amount of characteristics, partly those used by these researchers and additional characteristics that we consider to be important for the distinction mentioned before. A new method based on evolutionary algorithms will be used to realise a selection of the most relevant characteristics and to obtain a classifier that will be capable of distinguishing the different types of this pathology.

BACKGROUND

The electrocardiogram (ECG) is a diagnostic tool that measures and records the electrical activity of the heart
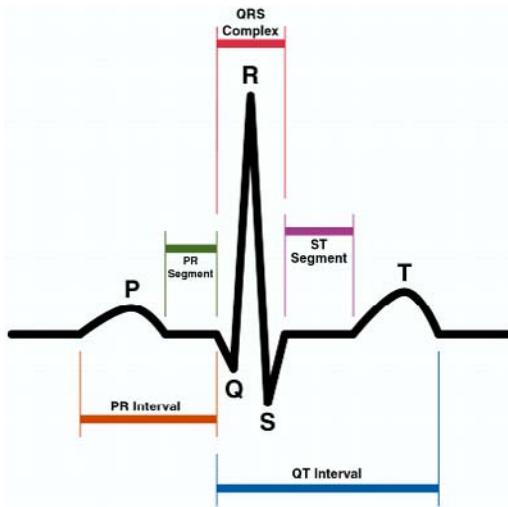
in exquisite detail (Lanza 2007). Interpretation of these details allows diagnosis of a wide range of heart conditions. The QRS complex is the most striking waveform within the electrocardiogram (Figure 1). Since it reflects the electrical activity within the heart during the ventricular contraction, the time of its occurrence as well as its shape provide much information about the current state of the heart. Due to its characteristic shape

it serves as the basis for the automated determination of the heart rate, as an entry point for classification schemes of the cardiac cycle, and often it is also used in ECG data compression algorithms.

A normal QRS complex is 0.06 to 0.10 sec (60 to 100 ms) in duration. In order to have a signal clean of auricular activity in the ECG, we will analyse and compare performances from these two different approaches:

1. To remove the activity of the QRS complex, subtracting from the signal a morphological average of its activity for every heart beat,
2. To detect the TQ section among heart beats (which are zones clean of ventricular activity) and analyse only data from that section.

There exists a great variety of algorithms to carry out the extraction of the auricular activity from the electrocardiogram such as the Thakor method (a recurrent adaptive filter structure), adaptive filtering of the whole band, methods based on neural-networks, spatial-temporal cancellation methods and methods based on Wavelets or on the concept of Principal Component Analysis (PCA) (Castells et al. 2004, Gilad-Bachrach et al. 2004, Petrutiu et al. 2004).

A fundamental step in any of these approaches is the detection of the QRS complex in every heart beat. Software QRS detection has been a research topic for

Figure 1. Diagram of the QRS complex

Figure 2. The segments are shown detected by the algorithm on the two channels of a registration. In green the end of the wave T is shown, and in red the principle of the wave Q. Therefore each tract among final of wave T (green) and wave principle Q (red), it is a segment of auricular activity. The QRST complex is automatically detected with good precision.
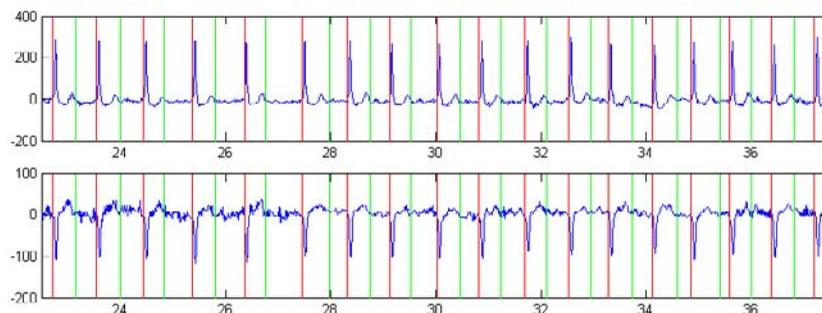
more than 30 years. Once the QRS complex is identified, we will have a starting point to implement some different techniques for the QRST removal. Figure 2 show how the QRST is automatically detected. This is the first step in the analysis of the ECG.

The study and analysis of feature extraction techniques from ECG signals is a very common task in any implementation of automatic classification systems from signals of any kind. During the execution of this sub-task, it is very important to analyse different research results existing in the literature.

It is important to analyse the use of the frequency domain to obtain the Dominant Atrial Frequency (DAF) which is an index of the auricular activity which measures the dominant frequency in the frequency spectrum that can be obtained from the auricular activity signal. In this spectrum, for each ECG record, the maximum energy peak is calculated, and this frequency will be the one that dominates the spectrum (Cantini et al. 2004). It is also important to use the RR distance, and different filters in the 4-10Hz range, using a Butterworth filter of first order. It is important to note the MUSIC (Multiple Signal Clasification) method of order 12 to calculate the pseudo-periodogram of the signal. In order to obtain more robust estimations, signal filtering by variable-length windows, with no overlapping, and on every one of them, an analysis of the frequency spectrum can be performed. It is also important to note the Welch method, the Choi-Williams transform, and some heuristical methods used by cardiology experts (Atrial Fibrillation, 2007).

GENETIC PROGRAMMING

 The genetic programming (GP) can be understood as an extension of the genetic algorithm (GA) (Zhao, 2007). GP began as an attempt to discover how computers could learn to solve problems, in different fields, like automatic desing, function approximation, classification, robotic control, signal processing, without being explicitly programmed to do so (Koza, 2003). Also, in bio-medical application, GP has been extensively and satisfactorily used (Lopes, 2007). The primary differences between GAS and GP can be summarised as a) GP typically codes solutions as tree structured, variable length chromosomes, while GA's generally make use of chromosomes of fixed length and structure, b) GP typically incorporates a domain specific syntax that governs acceptable (or meaningful) arrangements of information on the chromosome. For GA's, the chromosomes are typically syntax free.

The field of program induction, using a tree-structured approach, was first clearly defined by Koza (Koza, 2003). The following steps summarise the search procedure used with GP.

1. Create an initial population of programs, randomly generated as compositions of the function and terminal sets.
2. WHILE termination criterion not reached DO
    (a) Execute each program to obtain a performance (fitness) measure representing how well each program performs the specified task.
    (b) Use a fitness proportionate selection method to select programs for reproduction to the next generation.
    (c) Use probabilistic operators (crossover and mutation) to combine and modify components of the selected programs.
3. The fittest program represents a solution to the problem.

A NEW INTELLIGENT CLASSIFIER BASED ON GENETIC PROGRAMMING FOR ECG.

In the different articles we have studied, the authors did not use any algorithmic method in order to try to classify the electrocardiograms (Cantini et al. 2004, Lemay et al. 2004). The authors applied simple methods to try to establish the possible classification based on the classification capacity of one single characteristic or pairs of characteristics (through a graphic representation) (Hay et al. 2004). Nevertheless, the fact that one single characteristic might not be perfect individually to classify a group of patterns in the different categories, does not mean that combined with another or others it does not obtain some high percentages in the classification. Due to the great quantity of characteristics obtained from the ECG, a method to classify the patterns was needed, alongside a way of selecting the subgroup of characteristics optimal for classifying, since the great quantity of existing characteristics would introduce noise as soon as the search for the optimal classifier of the patterns of characteristics begins. In total 55 different

characteristics were used, from the papers (Cantini et al. 2004, Lemay et al. 2004, Hayn et al. 2004, Mora et al. 2004). There are other paper in the bibliography that used soft-computing method to analyze ECG (Wiggins et al. 2008, Lee et al. 2007, Yu et al. 2007).

In this paper, a new intelligent algorithm based on genetic programming (one paradigm of the soft-computing area) for simultaneously select the best features is proposed for the problem of classification spontaneous termination of atrial fibrillation. In this algorithm genetic programming is used to search for a good classifier at the same time as the search for an optimal subgroup of characteristics. The algorithm consists of a population of classifiers, and each one of those is associated with a fitness value that indicates how well it classifies. Each classifier is made up of:

1. A binary vector of characteristics, which indicates with 1's the characteristics it uses.
2. A multitree with as many trees as classes as has the collection of data of the problem. Every tree *i* distinguishes between the class *i* (giving a positive output) and the rest of the classes (negative output). Furthermore, it is connected to values p_j (frequency of failures), and wj (frequency of successes). The trees are made up of function nodes [+,-,*,/, trigonometric functions (sine, cosine, etc.), statistic functions (minimums, maximums, average)] and terminal nodes {constant number and features}. Their translation to a mathematical formula is immediate.

The algorithm consists of a loop in which in each repetition a new population is formed from the previous through the genetic operators. The classifiers that score the highest on fitness will have more possibilities to participate, with which the population will tend to improve its quality with the successive generations. The proposed algorithm is composed of the following building blocks:

1. **Fitness function**. The fitness function combines the double objective of achieving a good classification and a small subgroup of characteristics:

$$Fitness = f \bullet \left(1 + \alpha e^{-\frac{\beta}{n}}\right)$$

(1)

In this equation, f is the sum of the cases of success in the classification of the trees, β is the cardinality of the feature subset used, n is the total number of features and α is a parameter which determines the relative importance that we want to assign for correct classification and the size of the feature subset, calculated as:

$$\alpha = C\left(1 - \frac{gen}{TotalGen}\right)$$

(2)

where *C* is a constant, and *TotalGen* is the number of generations proposed genetic algorithm is evolved, and *gen* is the current generation number.

Figure 3. An example of a crossover operation in the proposed multitree classifier. (a) and (b) are initially the classifiers P1 and P2. In the figures (c) and (d) the results of the crossover operator is presented.

913
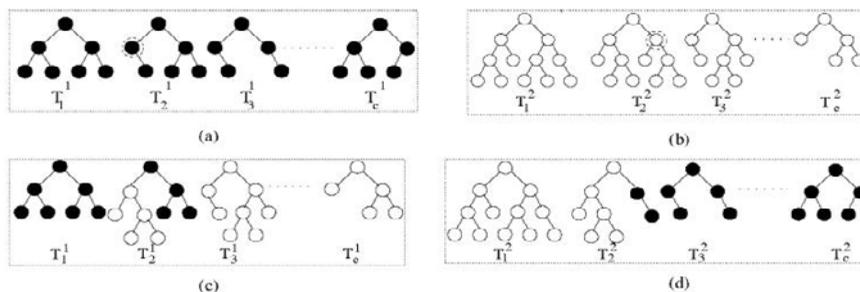
2. **Reproduction operator**: a classifier chosen proportionally to the fitness passes on, intact, to the next generation.
3. **Mutation operator**: a classifier is selected randomly and nodes of a tree are changed, giving more probability to the worst trees.
4. **Crossover operator**: homogeneous cross (classifiers with the same characteristics) and heterogeneous cross (classifiers with a similar subgroup). It realises the exchange of subtrees and trees between the classifiers. Figure 3 shows the behaviour of this operator.

It was thought to be useful to value the characteristics first, and use this assessment when a subgroup would be assigned to the classifier. This is performed in the following steps:

- A probability is given to each characteristic of being assigned to the initial subgroup of the classifier proportional to its assessment.
- G-flip was used to assess the characteristics (Gilad-Bacharach et al. 2004). G-flip is a greedy search algorithm for maximizing an evaluation function that take into account the number of features selected. The algorithm repeatedly iterates over the feature set and updates the set of chosen features. In each iteration it decides to remove or add the current feature to the selected set by evaluating the margin term of the evaluation function with and without this feature. This algorithm is similar

to the zero-temperature Monte-Carlo (Metropolis) method. It converges to a local maximum of the evaluation function, as each step increases its value and the number of possible feature sets is finite.

- The proposed methodology devalues bad characteristics in groups with a large quantity of characteristics, thus accelerating their convergence to good groups of characteristics and good classification results.

SIMULATION RESULTS

We have used and compared two different new intelligent classifiers. The first one presents an online feature selection algorithm using genetic programming. The proposed genetic programming methodology simultaneously selects a good subset of features and constructs a classifier using the selected features for the problem of ECG classification. We have designed new genetic operator in order to produce a robust and precise algorithm. The other classifier is based in the hybridization of a feature selection algorithm and a neural network system based on kernel method (Support Vector Machine).

We have four classification task:

✓ Event A: To differ among registration N (Group N: non-terminating AF -defined as AF that was not observed to have terminated for the duration

Table 1. Comparison of different approaches (in bracket the standard deviation)

Method:	Infogain (Molina et al. 2002)		New evolutive algorithm for classification		Kernel method (Support Vector Machine) (Schölkopf et al. 2002)		Relief (Kononenko 1994)	
Task	Best	Median/ (error)	Best	Median	Best	Median	Best	Median
Event A:	93	91 (±2)	100	98 (±2)	100	98 (±2)	72	64 (±8)
Event B:	70	66 (±4)	95	81 (±14)	80	68 (±12)	80	74 (±6)
Event C:	96	88 (±6)	89	83 (±6)	84	75 (±9)	74	68 (±6)
Event D:	68	62 (±4)	85	80 (±5)	83	77 (±6)	53	49 (±4)

of the long-term recording, at least an hour following the segment-) and registration T (Group T: AF that terminates immediately (within one second) after the end of the record).

✓ Event B: To differ among the type registrations S (Group S: AF that terminates one minute after the end of the record) and those of type T.

✓ Event C: To differ among registrations type N of AF and a second group in which registrations type S and type T are included.

✓ Event D: Separation of the 3 types of registrations in a simultaneous way.

These groups N,T and S are distributed across a learning set (consisting of 10 labelled records from each group) and two test sets. Test set A contains 30 records, of which about one-half are from group N, and of which the remainder are from group T. Test set B contains 20 records, 10 from each of groups S and T. Table 1 shows the simulation results (in % of classification), for different method and the evolutive algorithm proposed for ECG classification:

FUTURE TRENDS

The field of signal processing in bio-medical problems is an exciting and increasingly field nowadays. The rapid development of powerful microcomputers promoted the widespread application of software for electrocardiogram analysis and QRS detection algorithms in cardiological devices, and automatic classifiers.

However, and important research field for the next year, will be the hybridization of new intelligent techniques, as genetic algorithm and genetic programming, or other paradigms from soft-computing (fuzzy logic, neural networks, SVM, etc.), that improve the behaviour of standard classification algorithm for the diagnosis of different cardiological pathologies.

CONCLUSIONS

In this paper, a new online feature selection algorithm using genetic programming technique has been proposed as classifier for classification spontaneous termination of atrial fibrillation. In a combined way, our genetic programming methodology automatically selects the required features while design the multitree classifier.

Different genetic operator has been design for the multitree classifier, and for a better performance of the classifier, the initialization process generates solution using smaller feature subsets with has been previously selected with a greedy search algorithm (G-Flips) for maximizing the evaluation function. The effectiveness of the proposed scheme is demonstrated in a real problem: The Classification Spontaneous Termination of Atrial Fibrillation. At this point, it is important to note that the use of different characteristic gives different classification result as can be observed by the authors working in this challenge. The selection of different features extracted from an electrocardiogram has a strong influence on the problem to be solve and in the behaviour of the classifier. Therefore it is important to develop a general tool able to be face with different cardiac illnesses, which can select the most appropriate features in order to obtain an automatic classifier. As it can be observed, the proposed methodology has very good result compared with the winner of the challenge from PhysioNet and Computers in Cardiology 2004, even if this methodology has been developed in a general way to resolved different classification problems.

REFERENCES

Atrial Fibrillation Atrial Flutter Fibrillation- What are they? http://www.hoslink.com/heart/af.htm).

Cantini, F., et al. (2004). Predicting the end of an Atrial Fibrillation Episode: The PhysioNet Challenge, *Computers in Cardiology*, 121-124

Castells, F., Rieta J.J., Mora C., Millet J., & Sánchez C. (2004). Estimation of Atrial Fibrilatory Waves fron one-lead ECGs using principal component analysis concepts, *Computer in Cardiology*, 215-219.

Gilad-Bachrach, R., Navot, A., & Tishby, N. (2004). Margin based feature selection - theory and algorithms. *ICML*, 4-8 .

Hayn, D., et al. (2004). Automated Prediction of Spontaneous Termination of Atrial Fibrillation from Electrocardiograms. *Computers in Cardiology*, 117-120

Kononenko, I. (1994), Estimating Attributes: Analysis and Extensions of RELIEF. *ECML,* 171–182.

Koza, J.R., et.al. (2003).*Genetic Programming IV: Routine Human-Competitive Machine Intelligence.* Kluwer Academic Publishers,

Lanza, G.A., (2007). The Electrocardiogram as a Prognostic Tool for Predicting Major Cardiac Events. *Cardiovascular Diseases*,(50) 2, 87-111

Lee., C.S., & Wang, M.H. (2007) Ontological fuzzy agent for electrocardiogram application. *Expert Systems with Applications*, Available online

Lemay, H., et al. (2004). AF Classification Based on Clinical Features. *Computers in Cardiology*, 669-672

Lopes, H.S., (2007).Genetic programming for epileptic pattern recognition in electroencephalographic signals. *Applied Soft Computing*, (7) 1, 343-352

Molina, L., Belanche, L., & Nebot A. (2002). Feature Selection Algorithms: A Survey and Experimental Evaluation, *IEEE International Conference on Data Mining*, 306-313

Mora, C., & Castells., J. (2004). Prediction of Spontaneous Termination of Atrial Fibrillation Using Time Frecuency Analysis of the Atrial Fibrillation Wave, *Computers in Cardiology*,109-112

Petrutiu, S., Sahakian, A.V., & Swiryn, S. (2004). Fibrillatory Wave Analysis of the Surface ECG to Predict Termination of Atrial Fibrillation. *Computers in Cardiology,* 250-261

Schölkopf, B. & Smola, A.J.(2002). *Learning with Kernels*. MIT Press, Cambridge, MA.

Wiggins, M., Saad, A., Litt, B., & Vachtsevanos, G., (2008). Evolving a Bayesian classifier for ECG-based age classification in medical applications. *Applied Soft Computing*, (8) 1,599-608

Yu, S.N., & Chen, Y.H. (2007). Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network. *Pattern Recognition Letters*, (28) 10, 1142-1150

Zhao, H., (2007).A multi-objective genetic programming approach to developing Pareto optimal decision tres. *Decision Support Systems*, (43) 3, 809-826

KEY TERMS

Arrhythmia: Arrhythmias are disorders of the regular rhythmic beating of the heart. Arrhythmias can be divided into two categories: ventricular and supraventricular.

Atrial Fibrillation: The atrial fibrillation (AF) is the sustained arrhythmia that is most frequently found in clinical practice, present in 0.4% of the total population. Its frequency increases with age and with the presence of structural cardiopathology. AF is especially prevalent in the elderly, affecting 2-5% of the population older than 60 years and 10 percent of people older than 80 years.

Electrocardiogram: The electrocardiogram (ECG) is a diagnostic tool that measures and records the electrical activity of the heart

Feature Selection: *Feature selection* is a process frequently used in classification algorithm, wherein a subset of the features available from the data are selected for the classifier. The best subset contains the least number of dimensions or features that most contribute to a correct classification process.

Genetic Algorithm: Genetic Algorithms (GA) are a way of solving problems by mimicking the same processes mother nature uses. They use the same combination of selection, recombination and mutation to evolve a solution to a problem.

Genetic Programming: Genetic Programming (GP), evolved a solution in the form of a Lisp program using an evolutionary, population-based, search algorithm which extended the fixed-length concepts of genetic algorithms.

Soft-Computing: Refers to a collection of different paradigms (such as fuzzy logic, neural networks, simulated annealing, genetic algorithms and other computational techniques), which are focussed in analyze, model and discover information in very complex problems.

Support Vector Machine (SVM): Are a special Neural Networks that performs classification by constructing an N-dimensional hyperplane that separates the data into two categories.

Intelligent MAS in System Engineering and Robotics

G. Nicolás Marichal
University of La Laguna, Spain

Evelio J. González
University of La Laguna, Spain

INTRODUCTION

The concept of agent has been successfully used in a wide range of applications such as Robotics, e-commerce, agent-assisted user training, military transport or health-care. The origin of this concept can be located in 1977, when Carl Hewitt proposed the idea of an interactive object called *actor*. This actor was defined as a computational agent, which has a mail address and a behaviour (Hewitt, 1977). Actors receive messages from other actors and carry out their tasks in a concurrent way.

It is difficult that a single agent could be sufficient to carry out a relatively complex task. The usual approach consists of a society of agents - called Multiagent Systems (MAS) -, which communicate and collaborate among them and they are coordinated when pursuing a goal.

The purpose of this chapter is to analyze the aspects related to the application of MAS to System Engineering and Robotics, focusing on those approaches that combine MAS with other Artificial Intelligence (AI) techniques.

BACKGROUND

There is not an academic definition accepted by every researcher about the term agent. In fact, agent researchers have offered a variety of definitions explicating his or her particular use of the word. An extensive list of these definitions can be found in (Franklin and Graesser, 1996). It does not fall in the scope of this chapter to reproduce that list. However, we will include some of them, in order to illustrate how heterogeneous these definitions are.

"Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed." (Maes, 1995, p. 108)

"Autonomous agents are systems capable of autonomous, purposeful action in the real world." (Brustoloni, 1991, p. 265)

"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors." (Russell and Norvig, 1995, p. 31)

Despite the existing plethora of definitions, agents are often characterized by only describing their features (long-live, autonomy, reactivity, proactivity, collaboration, ability to perform in a dynamic and unpredictable environment, etc.). With these characteristics, users can delegate to agents tasks designed to be carried out without human intervention, for instance, as personal assistants that learn from its user.

In most of applications, a standalone agent is not sufficient for carrying out the desired task: agents are forced to interact with other agents, forming a MAS. Due to their capacity of flexible autonomous action, MAS can treat with open – or at least highly dynamic or uncertain- environments. On the other hand, MAS can effectively manage situations where distributed systems are needed: the problem being solved is itself distributed, the data are geographically distributed, systems with many components and huge content, systems with future extensions, etc. A researcher could include a single agent to implement all the tasks. Nevertheless, this type of macroagent represents a bottleneck for the system speed, reliability and management.

It is clear that the design of a MAS is more complex than a single agent. Apart from the code for the treatment of the task-problem, a developer needs to implement those aspects related to communication, negotiation among the agents and its organization in the system. Nevertheless, it has been shown that MAS offer more than they cost (Cockburn, 1996) (Gonzalez, 2006) (Gonzalez, 2006b) (Gyurjyan, 2003) (Seilonen 2005).

MAS, AI AND SYSTEM ENGINEERING

An important topic in System Engineering is that of process control problem. We can define it as the one of manipulating the input variables of a dynamic system in an attempt to influence over the output variables in a desired fashion, for example, to achieve certain values or certain rates (Jacquot, 1981). In this context, as other Engineering disciplines, we can find a lot of relevant formalisms and standards, whose descriptions are out of the scope of this chapter. An interested reader can get an introductory presentation of these aspects in (Jacquot, 1981).

Despite their advantages, there are few approaches to the application of MAS technology to process automation (much less than applications to other fields such as manufacturing industry). Some reasons for this lack of application can be found in (Seilonen, 2005):

- Process automation requires run-time specifications that are difficult to reach by the current agent technology.
- The parameters in the automation process design are usually interconnected in a strict way, thus it is highly difficult to decompose the task into agent behaviors.
- Lack of parallelism to be modeled through agents.

In spite of these difficulties, some significant approaches to the application of MAS to process control can be distinguished:

- An interesting approach of application of MAS to process control is that in which communication techniques among agents are used as a mechanism of integration among systems independently designed. An example of this approach is the

ARCHON (*Architecture for Cooperative Heterogeneous on-line systems*) architecture (Cockburn, 1996) that has been used in at least three engineering domains: Electricity Transportation, Electricity Distribution and Particle Accelerator Control. In ARCHON, each application program (known as *Intelligent System*) is provided with a layer (called *Archon Layer*) that allows it to transfer data/messages to other Intelligent Systems.

- A second approach consists of those systems that implement a closed loop-based control. In this sense, we will cite the work of (Velasco et al., 1996) for the control of a thermal central.
- A different proposal consists of complementing a pre-existing process automation system with agent technology. In other words, it is a complementation, not a replacement. The agent system is an additional layer that supervises the automation system and reconfigures it when it is necessary. Seilonen et al. also propose a specification of a BDI-model-based agent platform for process automation (Seilonen, 2005).
- V. Gyurjyan et al. (2003) propose a controller system architecture with the ability of combining heterogeneous processes and/or control systems in a homogeneous environment. This architecture (based on the FIPA standard) develops the agents as a level of abstraction and uses a description of the control system in a language called COOL (*Control Oriented Ontology Language*).
- Tetiker et al. (2006) propose a decentralized multi-layered agent structure for the control of distributed reactor networks where local control agents individually decide on their own objectives allowing the framework to achieve multiple local objectives concurrently at different parts of the network. On top of that layer, a global observer agent continuously monitors the system.
- Horling, Lesser et al. (2006) describe a soft real-time control architecture designed to address temporal and ordering constraints, shared resources and the lack of a complete and consistent world view. From challenges encountered in a real-time distributed sensor allocation environment, the system is able to generate schedules respecting temporal, structural and resource constraints, to merge new goals with existing ones, and to detect and handle unexpected results from activities. Other proposal of real-time control architecture

is CIRCA (*A Cooperative Intelligent Real-Time Control Architecture*) by Musliner, Durfee and Shin (1993), that uses separate AI and real-time subsystems to address the problems for which each is designed.

In this context, we proposed a MAS (called MAS-CONTROL) for identification and control of processes, whose design follows the FIPA specifications (FIPA, 2007) regarding architecture, communication and protocols. This MAS implements a self-tuning regulator (STR) scheme, so this is not a new general control algorithm but a new approach for its development. Its main contribution consists of showing the potential that a controller, through the use of MAS and ontologies – expressed in OWL (*Ontology Web Language)-*, can control systems in an autonomous way, using actions whose description, for example, is on the web, and can read on it (without knowing a priori) the logic of how to do the control. In this context, our experience is that agents do not offer any advantage if they are not intelligent and ontologies represent an intelligent way to manage knowledge since they provides the common format in which they can express that knowledge. Two important advantages of their use are extensibility and communication with other agents sharing the same language. These advantages are shown in the particular case of open systems, that is, when different MAS from different developers interact (Gonzalez, 2006).

As a STR, our MAS tries to carry out the processes of identification and control of a plant. We consider that this model can be properly managed by a MAS due to two main reasons:

- A STR scheme contains modules that are conceptually different, such as the direct interaction with the plant to control, identification of the system and determination of the best values for the controller parameters.
- It is possible to carry out the calculations in a parallel way. For instance, several transfer functions could be explored simultaneously. Thus, several agents can be launched in different computers, taking advantage of the possibility of parallelism provided by the MAS.

Other innovator aspect of this work is the use of artificial neural networks (ANN) for the identification and determination of the parameters. ANN and STR present clear analogies. The training of a neural network consists of finding the best values of the weights of the network while it is necessary to optimize some parameters for a model (identification) or for a controller in a STR. Because of this similarity of methods, we have considered the application of ANN training methods to control problems. In this case, ANN are applied for two purposes: the parameter optimization of a model of the unknown system and the optimization of the parameters of a controller. This way, the resulting system could be seen as a hybrid intelligent system for a real-time application. An interested reader can get a deeper description of the system in (Gonzalez, 2006b).

It is important to remark that this framework can be used for every algorithm of identification and control. In this context, we have checked the MAS controlling several and different plants, obtaining a proper behavior. In contrast, due to the transmission rate and optimization time, the designed MAS should be used for the controlling of not-excessively fast processes, according to the first restriction stated above. However, we expect to have shown an example of how the other two (strong interdependency of the parameters and lack of parallelism) can be overcome.

As can be seen, the mentioned restrictions often become serious obstacles in the application of MAS to Engineering Systems. In this framework, the use of Fuzzy rules is a very usual solution in order to define single-agents behaviours (Hoffmann, 2003). Unfortunately, the definition of the rules is cumbersome in most cases. As a possible solution to the difficult task of generating the adequate rules, several automatic algorithms have been proposed. New rule extraction approaches based on connectionist models have been proposed. Among them, the Neuro-Fuzzy systems has been proven as a way to obtain the rules, taking advantage of the learning properties of the Neural Networks and the form of expressing the knowledge by Fuzzy rules (Mitra and Hayashi, 2000).

In this context, several applications have been developed. In Robotics applications, it could be cited the work of (Lee and Qian, 1998), who describe a two-component system for picking up moving objects for a vibratory feeder or the work of (Kiguchi, 2004), proposing a hierarchical neuro-fuzzy controller for a robotic exoskeleton, to assist motion of physically weak persons such as elderly, disabled, and injured persons. As a particular case, a system for the detection and identification of road markings will be presented

in this chapter. This system has been incorporated to a vehicle as it can be seen in Figure 1.

This system is based on infrared technology and a classification tool based on a Neuro-Fuzzy System. A particular feature to take into account in this kind of tasks is that the detection and classification have to be done in real time. Hence, the time consumed by the hardware system and the processing algorithms is critical in order to take a right decision within the time frame of its relevance. Looking for an inexpensive and fast system, the infrared technology is a good alternative solution in this kind of applications. In this direction, taking into account the time limitations, a combination between a device based on infrared technology and different techniques to extract convenient Fuzzy rules are used (Marichal, 2006). It is important to remark that the extraction and the interpretation of the rules have generated great interest in recent years (Guillaume, 2007).

The final purpose is to achieve a MAS, where each agent does its work as fast as possible, overcoming the temporal limitations of the MAS as pointed out by (Seilonen, 2005). In this context, we would like to remark some approaches of MAS applied to decision fusion for distributed sensor systems, in particular that by Yu and Sycara (2006). In order to achieve the mentioned MAS, it is necessary to obtain the rules for each agent. Furthermore, a depth analysis over the rules has to be done, minimizing the number of them and setting the mapping between these rules and the different scenarios.

The approach used in the shown case is based on designing rules for each situation found by the vehicle. In fact, each different scenario should be expressed

Figure 1. Infrared system under the vehicle

Table 1. Rules extracted by the neuro-fuzzy approach

	Arrow	Right Arrow	Yield	Forward-right Arrow	Other Rules
Range	[0 2)	[2 4)	[4 6)	[6 8]	[-1 0]
Reference Value	1	3	5	7	
Rules	6, 7	8,9, 10,11, 12	13,14, 15,16, 17,18	19,20,21, 22,23, 24, 25	1,2, 3,4, 5

by its own rules. This feature gives more flexibility in the process of designing the desired MAS. Because of that, the separation of rules according to the kind of road marking could help in this purpose. In Table 1, it is shown the result of this process for the infrared system shown in Figure 1. Note that, the reference values are the values associated with each road marking, the range refers to the interval where the output values of the resultant Fuzzy system could be for a particular sign and finally, the rules are indicated by an order number.

It is important to remark that it is necessary to interpret the obtained rules. In this way, it is possible to associate these rules with different situations and generate new rules more appropriate for a particular case under consideration. Hence, the agents related with the detection and classification of the signs could be expressed by this set of Fuzzy rules. Moreover, agents, which are in charge of taking decisions based on the information, provided by the detection and classification of a particular road marking, could incorporate these rules as part of them. Problems in task decomposition process, pointed out by (Seilonen, 2005), could be simplified in this way. On the other hand, although the design of behaviors is very important, it should be said that the issues related with the co-operation among agents are also essential. In this context, the work of (Howard et al, 2007) could be cited.

FUTURE TRENDS

As technology provides faster and more efficient computers, the application of AI techniques to MAS is supposed to become increasingly popular. That improvement in the computer capacity and some emerging techniques (meta-level accounting, schedule caching, variable time granularities, etc.) (Horling, Lesser et al., 2006) will imply that other AI methods- impossible to be currently applied in the field of System Engineering- will be introduced in an efficient way in a near future.

In our opinion, other important feature to be explored is the improvement in MAS communication. It is also convenient to look for more efficient MAS protocols and standards, in addition to those aspects related to new hardware features. These improvements would allow, for example, developing operative real-time tele-operated applications.

CONCLUSION

The application of MAS to Engineering Systems and Robotics is an attractive platform for the convergence of various AI technologies. This chapter shows in a summarized manner how different AI techniques (ANN, Fuzzy rules, Neuro-Fuzzy systems) have been

successfully included into MAS technology in the field of System Engineering and Robotics. These techniques can also overcome some of the traditionally described drawbacks for MAS application, in particular, highly difficult decomposition of the task into agent behaviors and lack of parallelism to be modeled through agents.

However, present-day MAS technology does not fulfill completely the severe real-time requirements that are implicit in automation processes. Thus, and until the technology provides faster and more efficient computers, our opinion is that the application of AI techniques in MAS needs to be optimized for real-time systems, for example, extracting convenient Fuzzy rules and minimizing its number.

REFERENCES

Brustoloni, J.C. (1991). Autonomous Agents: Characterization and Requirements. *Carnegie Mellon Technical Report CMU-CS-91-204*, Pittsburgh: Carnegie Mellon University

Cockburn D. & Jennings, N. R. (1996). ARCHON: A Distributed Artificial Intelligence System for Industrial Applications. In G.M.P. O'Hare and N.R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons.

Franklin S. & Graesser A. (1996). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Intelligent Agents III. Agent Theories, Architectures and Languages (ATAL'96)*.

FIPA web site. http://www.fipa.org. Last access: 15th August 2007.

González, E.J., Hamilton, A., Moreno, L., Marichal, R., & Muñoz V. (2006) Software experience when using ontologies in a multi-agent system for automated planning and scheduling. *Software - Practice and Experience*, 36 (7), 667-688.

González, E.J., Hamilton, A., Moreno, L., Marichal, R., Marichal, G.N., & Toledo J. (2006b) A MAS Implementation for System Identification and Process Control. *Asian Journal of Control*, 8 (4). 417-423.

Destercke S., Guillaume S. and Charnomordic B. (2007) Building an interpretable fuzzy rule base from data using orthogonal least squares- application to a depollution problem. *Fuzzy Sets and Systems.* 158, 2078-2094

Gyurjyan, V., Abbott, D., Heyes, G., Jastrzembski, E., Timmer, C. & Wolin, E. (2003) FIPA agent based network distributed control system. *2003 Computing in High Energy and Nuclear Physics (CHEP03)*.

Hewitt, C. (1977). Viewing Control structures as Patterns of Passing Messages. *Artificial Intelligence*, (8) 3, 323-364.

Hoffmann, F. (2003). An Overview on Soft Computing in Behavior Based Robotics. *Lecture Notes in Computer Science*, 544-551.

Horling, B., Lesser V., Vincent R. & Wagner T. (2006) The Soft Real-Time Agent Control Architecture, *Autonomous Agents and Multi-Agent Systems,* 12(1), 35-91

Howard A, Parker L. E., and Sukhatme G., (2006). Experiments with a Large Heterogeneous Mobile Robot Team: Exploration, Mapping, Deployment, and Detection. *International Journal of Robotics Research*, vol. 25, 5-6, 431-447.

Jacquot, R.G. (1981) *Modern Digital Control Systems*. Marcel Dekker, Editor. Electrical engineering and electronics; 11.

Kiguchi, K.; Tanaka, T.; Fukuda, T. (2004) Neuro-fuzzy control of a robotic exoskeleton with EMG signals, *IEEE Transactions on Fuzzy Systems* 12, 4, 481 - 490.

Lee, K. M. & Qian, Y. F. (1998) Intelligent vision-based part-feeding on dynamic pursuit of moving objects, *Journal Manufacturing Science Engineering–Transactions ASME* 120(3), 640–647.

Maes, P. (1995). Artificial Life Meets Entertainment: Life like Autonomous Agents, *Communications of the ACM*, 38 (11), 108-114

Marichal, G.N., González, E.J., Acosta, L., Toledo, J., Sigut, M. & Felipe, J. (2006). An Infrared and Neuro-Fuzzy-Based Approach for Identification and Classification of Road Markings. *Advances in Natural Computation. Lecture Notes in Computer Science*, 4.222, 918-927.

Mitra, S. & Hayashi, Y. (2000). Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks*. (11) 3, 748-768

Musliner, D., Durfee E. & Shin, K. (1993). CIRCA: A Cooperative Intelligent Real-Time Control Architecture, IEEE Transactions on Systems, Man and Cybernetics, 23(6)

Russell, S.J. & Norvig, P. (1995), *Artificial Intelligence: A Modern Approach,* Englewood Cliffs, NJ: Prentice Hall

Seilonen, I., Koskinen, K., Pirttioja, T., Appelqvist, P. & Halme, A. (2005). Reactive and Deliberative Control and Cooperation in Multi-Agent System Based Process Automation, *6th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2005)*.

Tetiker, M.D., Artel, A., Tatara, E., Teymour, F., North, M., Hood, C. & Cinar, A. (2006) Agent-based System for Reconfiguration of Distributed Chemical Reactor Network Operation, *Proceedings of the American Control Conference*.

Velasco, J., González, J.C., Magdalena, L. & Iglesias, C. (1996). Multiagent-based control systems: a hybrid approach to distributed process. *Control Engineering Practice*, 4, 839-846.

Yu B. & Sycara K. (2006) Learning the Quality of Sensor Data in Distributed Decision Fusion, *International Conference on Information Fusion (Fusion 06)*, Florence, Italy.

KEY TERMS

Artificial Neural Network: An organized set of many simple processors called *neurons* that imitates a biological neural configuration.

FIPA: It stands for "Foundation for Intelligent Physical Agents", IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies

MultiAgent System: System composed of several agents, usually designed to cooperate in order to reach a goal.

Neuro-Fuzzy: Hybrids of Artificial neural networks and Fuzzy Logic.

Ontology: Set of classes, relations, functions, etc. that represents knowledge of a particular domain.

Real-Time System: System with operational deadlines from event to system response.

Self-Tuning Regulator: Type of adaptive control system composed of two loops, an inner loop (process and ordinary linear feedback regulator), and an outer loop (recursive parameter estimator and design calculation which adjusts its parameters).

Intelligent Query Answering Mechanism in Multi Agent Systems

Safiye Turgay
Abant İzzet Baysal University, Turkey

Fahrettin Yaman
Abant İzzet Baysal University, Turkey

INTRODUCTION

The query answering system realizes the selection of the data, preparation, pattern discovering, and pattern development processes in an agent-based structure within the multi agent system, and it is designed to ensure communication between agents and an effective operation of agents within the multi agent system. The system is suggested in a way to process and evaluate fuzzy incomplete information by the use of fuzzy SQL query method. The modelled system gains the intelligent feature, thanks to the fuzzy approach and makes predictions about the future with the learning processing approach.

The operation mechanism of the system is a process in which the agents within the multi agent system filter and evaluate both the knowledge in databases and the knowledge received externally by the agents, considering certain criteria. The system uses two types of knowledge. The first one is the data existing in agent databases within the system and the latter is the data agents received from the outer world and not included in the evaluation criteria. Upon receiving data from the outer world, the agent primarily evaluates it in knowledgebase, and then evaluates it to be used in rule base and finally employs a certain evaluation process to rule bases in order to store the knowledge in task base. Meanwhile, the agent also completes the learning process.

This paper presents an intelligent query answering mechanism, a process in which the agents within the multi-agent system filter and evaluate both the knowledge in databases and the knowledge received externally by the agents. The following sections include some necessary literature review and the query answering approach Then follow the future trends and the conclusion.

BACKGROUND

The query answering system in agents utilizes fuzzy SQL queries from the agents, then creates and optimizes a query plan that involves the multiple data source of the whole multi agent system. Accordingly, it controls the execution of the task to generate the data set. The query operation constitutes the basic function of query answering. By query operation, the most important function of the system is fulfilled. This study also discusses peer to peer network structure and SQL structure, as well as query operation.

Query operation was applied in various fields. For example, selecting the related knowledge in a web environment was evaluated in terms of relational concept in databases. Relational database system particularly assists the system in making evaluations for making decisions about the future and in making the right decisions with fuzzy logic approach (Raschia & Mauaddib, 2002; Tatarinov et al. 2003; Galindo et al. 2001; Bosc et al. Chaudhry et.al. 1999; Saygın et al. 1999; Turgay et al.2006).

Query operation was mostly used in choosing the related information web environment (Jim & Suciu, 2001; He et al. (2004). Data mining approach was used in dynamic site discovery process by the data preparation and type recognition approaches in complex matching schema with correlation values in query interfaces and query schemas (Nambiar & Kambhampati, 2006; Necib & Freytag, 2005). Query processing within peer to peer network structure with SQL structure was discussed generally (Cybenko et al. 2004; Bernstein et al. 1981). Query processing and database was reviewed with relational database (Genet & Hinze, 2004; Halashek-Wiener et al., 2006). Fuzzy set was proposed by Zadeh (1965) and the division of the features into various linguistic values was widely

used in pattern recognition and in the fuzzy inference system. Kubat, et al. (2004) reviewed the frequency of the fuzzy logic approach in operations research methods as well as artificial intelligence ones in discrete manufacturing. Data processing process within the multi-agent systems can be grouped as static and dynamic. While the evaluation process of existing data by the system can be referred to as a static structure, the evaluation process of new data or possible data within the system can be referred to as a dynamic structure. The studies on the static structure can be expressed as database management's query process (McClean, Scotney, Rutjes & Hartkamp, 2003) and the studies on the dynamic structure can be expressed as the whole of the agent system (Purvia, Cranefield, Bush & Carter, 2000; Hoschek, 2002; Doherty, Lukaszewicz, & Szalas, 2004, Turgay, 2006)

AGENT BASED QUERY ANSWERING SYSTEM

The query process lists the knowledge with desired characteristics in compliance with the required condition while query answering finds the knowledge conforming to the required conditions and responds to the related message in the form of knowledge. In par-

ticular, a well-defined query answering process within multi agent systems provides communication among agents, the sharing of knowledge and the effective performance of data processing process and learning activities. The system is able to process incomplete or fuzzy knowledge intelligently with the fuzzy SQL query approach.

The distributed query answering mechanism was proposed as a cooperative agent-based solution for information management with fuzzy SQL query. A multi-agent approach to information management includes some features such as:

- Concurrency
- Distributed computation
- Modularity
- Cooperation

Figure 1 represents each agent's query answering mechanism. When the data is received by the system, the query variables are chosen by query and then the data related with fuzzy SQL are suggested. The obtained result is represented as the answer knowledge in the agent and thus the process is completed.

The data are classified by the fuzzy query approach, depending on fuzzy relations and importance levels. The rule base of the system is formed after a

Figure 1. Model driven framework for query answering mechanism in a multi-agent system
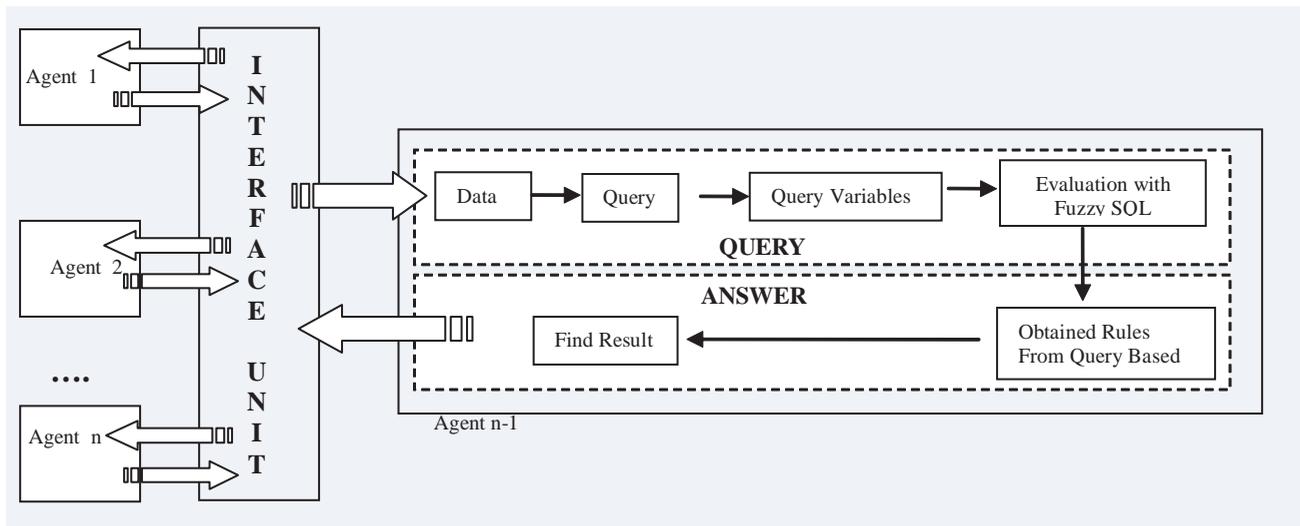
query and evaluation. The task base structure of the system is updated by the mechanism in line with the obtained fuzzy rules, and then, it is ensured that the system makes an appropriate and right decision and acts intelligently.

Operation Mechanism of Agent Based Fuzzy Query Answering System

The agent does the following:

Step1: receives the task knowledge from the related agent
Step2: does the fuzzification of knowledge
Step3: determines fuzzy grade values according to knowledge features

Step4: determines the knowledge in compliance with the criteria through fuzzy SQL commands
Step5: sends the obtained task or rule to the related agent
Step6: performs the answering operation

The agent based query answering system involves three main stages: knowledge processing, query processing and agent learning (see Figure2). The operation types of these stages are given in detail below.

Knowledge Processing

This is the stage where the knowledge is received by the agent from the external environment and necessary preparations are made before query. The criteria and

Figure 2. Suggested system model for each agent
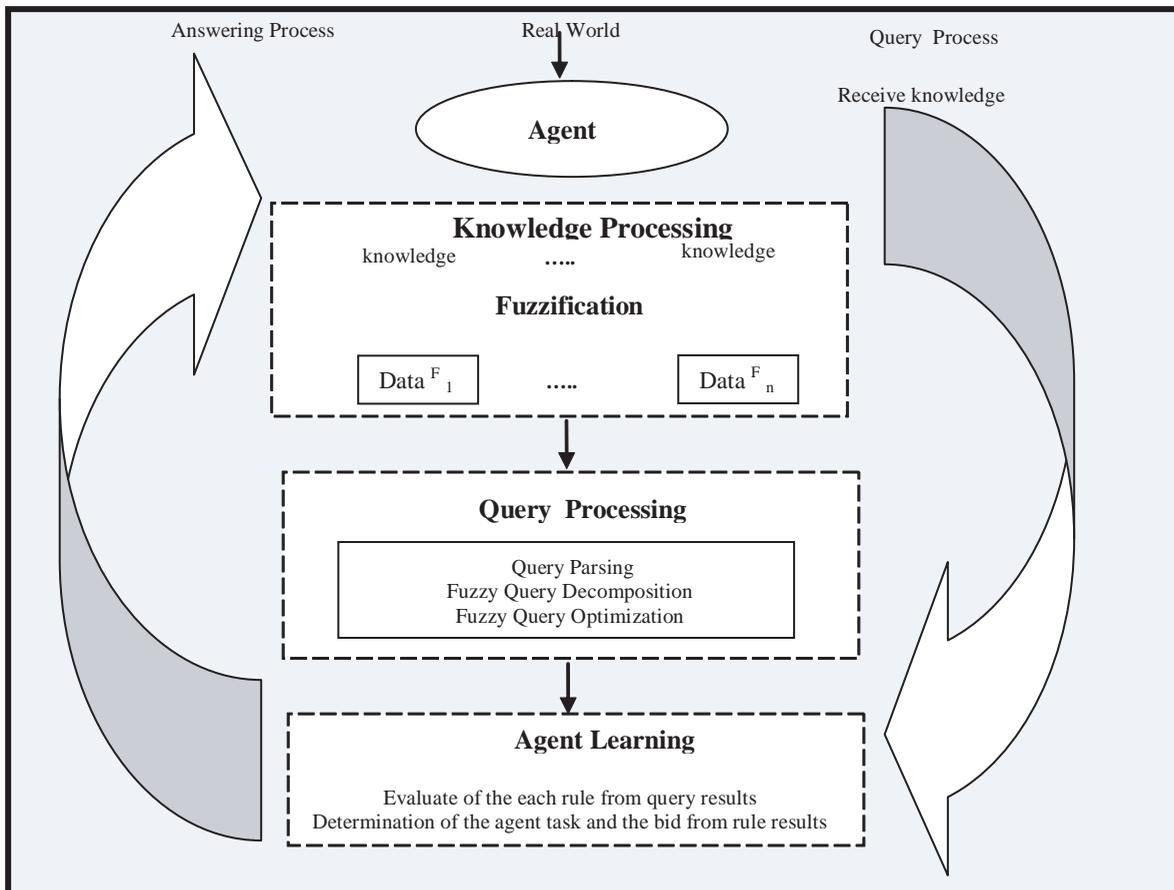
keywords to be used in evaluating the received data are defined in this stage. This stage can also be called pre-query. The keywords, concepts, attribute and relationship knowledge to be analysed by the agent are determined in this stage before query.

In this system, the behaviour structure of intelligent query answering system is formed. During the system modelling, the perception model considered being coming signal, data and knowledge from the external environment for a more understandable structure in learning module plays an important role. Coming from the external environment and called the input modelling; $<A_{i,x}, A_i, \varnothing>$ is defined as the perception set. Agent i, x perception coming from the external environment, refer to the $A_{i,x}$. Table 1 includes the nomenclature of agent based query answering system. The multi-agent system consists of more than one agent. The agent set is $A=\{A_1, A_2,…,A_i\}$. The knowledge set is $K=\{K_1, K_2, …,K_y\}$ The knowledgebase is <Definition of Knowledge, Attribute, Dependency Situation, Agent > (in Table 1 and Figure 3).

The rule set is $R=\{R_1, R_2, ...,R_x\}$. The rule base is <Definition of Rule, Attribute, Dependency Situation, Agent > . The task set is $T=\{T_1, T_2, ...,T_j\}$. The task base is <Definition of Task, Attribute, Dependency Situation, Agent > .

When data arrives from the external environment, it is perceived as input : $<A_{i,x}, A_i, \varnothing>$ When "x" is perceived by Agent i, it is referred to as $A_{ix.}$ This input can also be used in knowledgebase, rule base and task base. **The following goals that were determined as a result of the process and the evaluation of the information coming to the knowledge-base should have been achieved in the mechanism of intelligent query answering.**

- Goal definition
- Data selection
- Data preparation

Query Processing

The agent performs two types of query in the process of defining keywords, concepts or attributes during knowledge processing. The first is external query, which is realized among the agents, while the second is the internal query, where the agent scans the knowledge within itself. During these query processes, the fuzzy SQL approach is applied.

Feature-Attribute At and relation Re are elements formed among the components within the system. These elements are the databases of knowledgebase, rule base and task base. While attribute refers to agent specifications, Resource includes not only raw data externally received but also knowledgebase, rule base and task base which each agent possesses.

Table 1. The nomenclature of agent based query answering system

A	$\rightarrow$ i agent set $\{A_1, A_2,...,A_i\}$
T	$\rightarrow$ j task set in $\{T_1, T_2, ...,T_j\}$
$A_{i,x}$	$\rightarrow$ i agents x percept
$\bigcup_{k=1}^{m} T_{jk}$	$\rightarrow$ i agent's j task sets refers to continuing subsets from k to m situation
$L_{i,m}$	$\rightarrow$ i agents m learning situation
$Q_{i,n}$	$\rightarrow$ i agents n querying situation
At_i	$\rightarrow$ i agents attribute situation
$R_{i,r}$	$\rightarrow$ i agent's r decision situation
$K_{i,y}$	$\rightarrow$ i agent's y knowledgebase
$R_{i,x}$	$\rightarrow$ i agent's x rule base
$T_{i,t}$	$\rightarrow$ i agent's t task base

$A=\{At, Re(K_{i,y}, R_{i,x}, T_{i,t})\}$

Let P(At) denote the set of all possibility distributions that may be defined over the domain of an attribute At. A fuzzy relation R with $\cup$ schema $A_1, A_2, \ldots, A_n$, where A_i is an attribute is defined as $R=P(At_1) \times P(At_2) \times \ldots \times P(At_n) \times D$, where D is a system-supplied attribute for membership degree with a domain [0,1] and $\times$ denotes the cross product.

Each data value V of the attribute is associated with a possibility distribution defined over the domain of the attribute and has a membership function denoted by $\mu_v(x)$. If the data value is crisp, its possibility distribution is defined by

$$\mu_v(x) = \begin{cases} 1 & if \quad x = v \\ 0 & otherwise \end{cases} \quad (1)$$

Like standard SQL, queries in fuzzy SQL are specified in select statement of the following form:

SELECT   Attributes
FROM   Relations
WHERE   Selection Conditions.

The semantics of a fuzzy SQL query is defined based on satisfaction degrees of query conditions. Consider a predicate $X\Theta Y$ in a WHERE clause. The satisfaction degree, denoted by $d(X\Theta Y)$, is evaluated for values of X and Y. Let the value of X be v_1 and that of Y of v_2. Then,

$$d(X\Theta Y) = \max_{X,Y} \quad (\min(\mu_{v1}(X), \mu_{v2}(Y), \mu_\Theta(X,Y)) \quad (2)$$

where X and Y are crisp values in the common domain over which v_1 and v_2 are defined(Yang et al., 2001). Function Θ is a function that compares the degrees in terms of satisfaction among the variables. When the satisfaction degree is evaluated for X and Y the former takes the value of v_1, while the latter takes the value of v_2.

As shown in Figure 2, bids are taken as a set, the frequencies of the received bids are fixed and then the bids are decomposed into groups. The decomposed bids are included into databases of the multi-agent system. The information in databases is fuzzified and the interrelation between them is determined in terms of weight and importance level.

Agent Learning Process

This is a process where the system learns the knowledge obtained as a result of query as a rule or task. The system fulfils not only the task but also the learning process (in Figure 3). Learning process is acquired and the data from the external transition is processed by the agent system of the defined aim during the activities. Learning algorithm shows the variability of the system status(in Table 2).

In the learning process with the help of the query processing, candidate rules are determined by taking the fuzzy dimension attributes and the attribute measures into consideration. Therefore, it would be true to say that a hierarchical order from knowledge-base to rule-base and from rule-base to task-base is available in the system.

Algorithm Learning Analysis

Input: A relational view that contains a set of records and the questions for influence analysis.

Output: An efficient association rule.

Step1: Specifies the fuzzy dimension attribute and the measure attribute.

Step2: Identifies the fuzzy dimension item sets and calculates the support coefficient

Step3: Identifies the measure item sets and calculates the support coefficient.

Step4: Constructs sets of candidate rules, and computes the confidence and aggregate value.

Step5: Obtains a rule at the granularity level with greatest confidence, and forms a rule at the aggregation level with largest abstract value of the measure attribute.

Step6: Computes the assertions at different levels, exits if comparable (i.e., there is no inconsistency found in semantics at different levels).
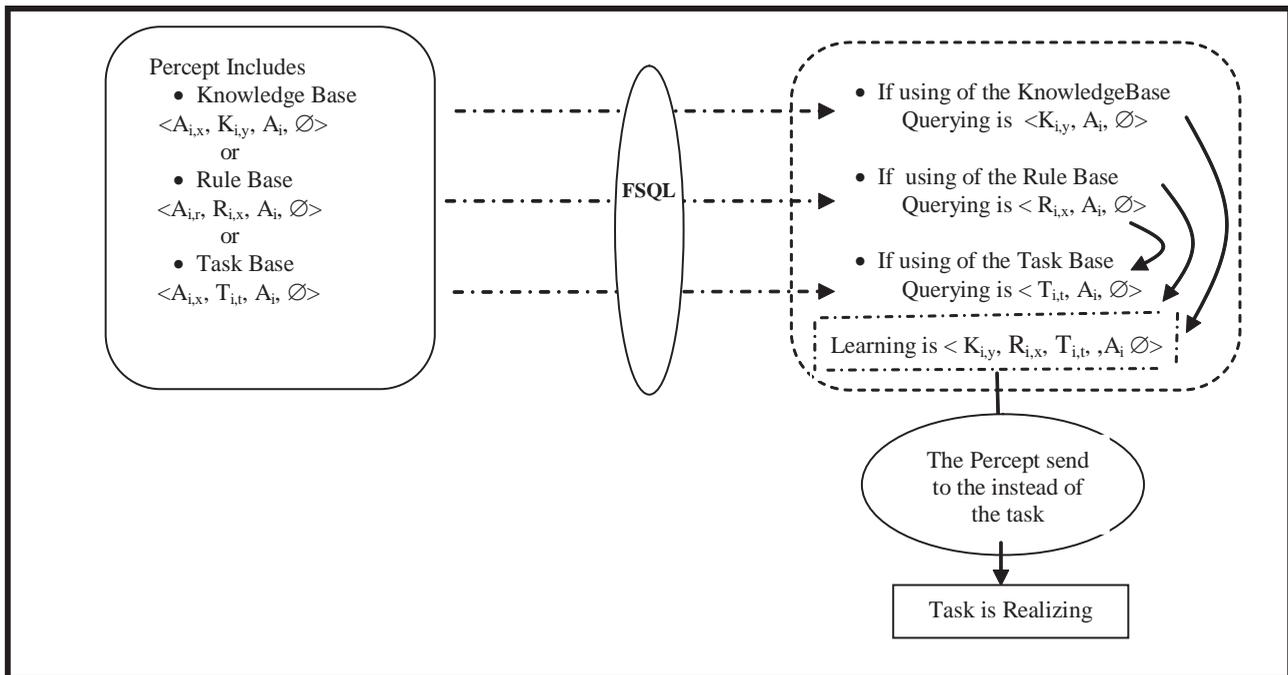
Step7: Generates rules from the refined measure item sets and forms the framework of the rule.

Step8: Constructs the final rule as a task for related agent.

Table 2. The query answering mechanism's learning analysis algorithm

Algorithm Learning Analysis

Input: A relational view that contains a set of records and the questions for influence analysis.

Output: An efficient association rule.

Step1: Specifies the fuzzy dimension attribute and the measure attribute.

Step2: Identifies the fuzzy dimension item sets and calculates the support coefficient

Step3: Identifies the measure item sets and calculates the support coefficient.

Step4: Constructs sets of candidate rules, and computes the confidence and aggregate value.

Step5: Obtains a rule at the granularity level with greatest confidence, and forms a rule at the aggregation level with largest abstract value of the measure attribute.

Step6: Computes the assertions at different levels, exits if comparable (i.e., there is no inconsistency found in semantics at different levels).

Step7: Generates rules from the refined measure item sets and forms the framework of the rule.

Step8: Constructs the final rule as a task for related agent.

Figure 3. The way the input perceived by the agent is processed

FUTURE TRENDS

Future tasks of the system will be realized when the system performs query answering more quickly thanks to the distributed, autonomous, intelligent and communicative agent structure of the suggested agent based fuzzy query answering system. In fuzzy approach, the system will primarily examine and group the relational database in databases of the agents with the fuzzy logic and then will shape the rule base of the system by applying the fuzzy logic method to these data. After the related rule is chosen, the rule base of the system will be designed and the decision mechanism of the system will operate. Therefore, relational database structure and system behaviour are important in determining the first peculiarity of the system and in terms of data clearing.

For future research, it is noted that the design of fuzzy databases involves not just modelling the data but also modelling operations on the data. Relational databases support only limited data types, while fuzzy and possibility databases allow a much larger number of comparatively complex data types (e.g., possibility distributions). This suggests that it might be fruitful to employ object-oriented database technology to allow explicit modelling of complex data types.

The incorporation of fuzziness into distributed events can be performed as a future study. Finally, due to frequent changes in the positions and status of objects in an active mobile database environment, the issue of temporality should be considered by adapting the research results of temporal database systems area into active mobile databases.

CONCLUSION

This paper discusses a variety of issues in adapting fuzzy database concepts to an active multi agent database system which incorporates active rules in a multi computing environment. This study shows how fuzziness can be introduced to different aspects of rule execution from event detection to coupling modes. As an initial step, membership degree calculation for various types of composite events has been explained. Dynamic determination of coupling modes has been done by using the strengths of events and reliabilities of conditions which are calculated via membership functions. Strengths of events and condition reliabili-

ties have been shown to be useful for condition and action status, as well. The partitioning of the rule set into multi agent system events has also been discussed as an example of inter-rule fuzziness. Similarity based event detection has been introduced to active multi agent databases, which is an important contribution from the perspective of performance.

REFERENCES

Bernstein, P.A., Goodman, N., Wong, E., Reeve, C.L. & Rothnie, J.B. (December 1981), Query Processing in a System for Distributed Databases (SDD-1), *ACM Transactions on Database Systems*, 6(4), 602-625.

Bosc, P. & Pivert, O. (1995), SQLf: A relational database language for fuzzy querying, IEEE Transactions on Fuzzy Systems, 3, 11-17.

Chaudhry, N., Moyne, J. & Rundensteiner, E.A. (1999), An extended database design methodology for uncertain data management", *Information Sciences*, 121, 83-112.

Cybenko, G., Berk, V., Crespi, V., Gray, R. & Jiang, G. (2004), An overview of Process Query Systems, *Proceedings of SPIE Defense and Security Symposium*, 12-16 April, Orlando, Florida, USA.

Doherthy, P., Szalas ,A. & Lukaszewicz, W. (2004), Approximate Databases and Query Techniques for Agents with Heterogeneous Ontologies and Perceptive Capabilities*, Proceedings on the 9th International Conference on Principles of Knowledge Representation and Reasoning.*

Doherty, P., Lukaszewicz, W. & Szalas, A. (2004), Approximate Databases and Query Techniques for Agents with Heterogeneous Perceptual Capabilities, *Proceedings on the 7th International Conference on Information Fusion*.

Doherty, Lukaszewicz, & Szalas, 2004

Galindo, J., Medina, J.M. & Aranda-Garrido, M.C. (2001), Fuzzy division in fuzzy relational databases: an approach, *Fuzzy Sets and Systems*, 121, 471-490.

Genet, B. & Hinze, A. (2004), Open Issues in Semantic Query Optimization in Related DBMS, *IV. Working paper series (University of Waikato. Dept. of Computer Science); 2004/10.*

Halashek-Wiener, C., Parsia, B. & Sinn, E. (2006), Towards Continuous Query Answering on the Semantic Web, *In UMIACS Technical Report*,. http://www.mindswap.org/papers/2006/ ContQueryTR2006.pdf.

He, B., Chang, K.C. & Han, J. (2004), Discovering Complex Matching across Web Query Interfaces: A Correlation Mining Approach, *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD'04, August 22-25, Seattle, Washington, USA.

Hoschek, W. (2002), Query Processing in Containers Hosting Virtual Peer-to-Peer Nodes, *Int'l. Conf. on Information Systems and Databases (ISDB 2002)*, Tokyo, Japan, September.

Jim, T. & Suciu D.(2001), Dynamically Distributed Query Evaluation, *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, Santa Barbara, California, United States, Pg: 28 – 39, 2001ISBN:1-58113-361-8

Kubat, C., Taşkın, H., Topal, B. & Turgay, S. (2004), Comparison of OR and AI methods in discrete manufacturing using fuzzy logic, *Journal of Intelligent Manufacturing*, 15, 517-526.

McClean, S., Scotney, B., Rutjes, H. & Hartkamp, J.,(2003), Metadata with a MISSION: Using Metadata to Query Distributed Statistical Meta-Information Systems, *2003 Dublin Core Conference: Supporting Communities of Discourse and Practice-Metadata Research&Applications, DC-2003*, 28 September-2 October, Seattle, Washington USA.

Nambiar, U. & Kambhampati,S. (2006), Answering Imprecise Queries over Autonomous Web Databases, *Proceedings of the 22nd International Conference on ICDE '06 Data Engineering*, 03-07 April 2006.

Necib, C. B. & Freytag, J.C.(2005), Query Processing Using Ontologies, *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAISE'05)*, Porto, Portugal, June.

Purvia, M., Cranefield, S., Bush, G. & Carter, D., (January 4-7, 2000), The NZDIS Project: an Agent-Based Distributed Information Systems Architecture, *in Proceedings of the Hawaii International Conference on System Sciences,* Maui, Hawaii.

Raschia, G. & Mouaddib, N. (2002), SAINTETIQ: a fuzzy set-based approach to database summarization, *Fuzzy Sets and Systems*, 129, 137-162.

Saygın, Y., Ulusoy Ö. & Yazıcı, A. (1999), Dealing with fuzziness in active mobile database systems, *Information Sciences*, 120, 23-44.

Tatarinov, I., Ives, Z., Madhavan, J., Halevy A., Suciu, D., Dalvi, N., Dong, X.L., Kadiyska, Y., Miklau & G., Mork, P. (September 2003), The Piazza Peer Data Management Project, *SPECIAL ISSUE: Special topic section on peer to peer data management ACM SIGMOD Record*, 32(3), , p 47 – 52, ISSN:0163-5808.

Turgay, S. (2006, May 29-31), Analytic Model of an Intelligent Agent Interface, *Proceedings of 5th International Symposium on Intelligent Manufacturing Systems*, Turkey, (pp.1222-1229).

Turgay, S., Kubat, C. & Öztemel, E. (2006, May 29-31), Intelligent Fuzzy Decision Mechanism in Multi-Agent Systems for Flexible Manufacturing Environment, *Proceedings of 5th International Symposium on Intelligent Manufacturing Systems*, Turkey, (pp. 1159-1169).

Yang, Q., Zhang, W., Liu, C., Wu, J., Yu, C., Nakajima, H & Rishe, N.D. (2001). Efficient Processing of Nested Fuzzy SQL Queries in a Fuzzy Database, *IEEE Transactions on Knowledge and Data Engineering*, 13(6).

Zadeh, L.A. (1965), Fuzzy sets, *Information Control*, 8 (3), 338–353.

KEY TERMS

Agent : A system that fulfils the independent functions, perceives the outer world and establishes the linking among the agents through its software.

Flexible Query: Incorporates some elements of the natural language so as to make a possible simple and powerful expression of subjective information needs.

Fuzzy SQL(Structural Query Language): It is an extension of the SQL language that allows us to write flexible conditions in our queries. The FSQL allows us to use linguistic labels defined on any attribute.

Fuzzy SQL Query: Fuzzy SQL allows the system to make flexible queries about crisp or fuzzy attributes in fuzzy relational data or knowledge.

Intelligent Agent: It consists of a sophisticated intelligent computer program; which is acting of situated, independent, reactive, proactive, flexible, recovers from failure and interacts with other agents.

Multi-Agent System: It is a flexible incorporated network of software agents that interact to solve the problems that are beyond the individual capacities or knowledge of each problem solver.

Query: Caries out the scanning of the data with required specifications.

Query Answering: Answers a user query with the help of a single or multi-database in the multi agent system.

System: A set of components considered to act as a single goal-oriented entity.

Intelligent Radar Detectors

Raúl Vicen Bueno
University of Alcalá, Spain

Manuel Rosa Zurera
University of Alcalá, Spain

María Pilar Jarabo Amores
University of Alcalá, Spain

Roberto Gil Pita
University of Alcalá, Spain

David de la Mata Moya
University of Alcalá, Spain

INTRODUCTION

The **Artificial Neural Network**s (**ANN**s) are based on the behaviour of the brain. So, they can be considered as **intelligent system**s. In this way, the **ANN**s are constructed according to a brain, including its main part: the neurons. Moreover, they are connected in order to interact each other to acquire the followed intelligence. And finally, as any brain, it needs having memory, which is achieved in this model with their weights.

So, starting from this point of view of the **ANN**s, we can affirm that these systems are able to learn difficult tasks. In this article, the task to learn is to distinguish between the presence or not of a reflected signal called **target** in a Radar environment dominated by **clutter**. The **clutter** involves all the signals reflected from other objects in a Radar environment that are not the desired **target**. Moreover, the **noise** is considered in this environment because it always exists in all the communications systems we can work with.

BACKGROUND

The **ANN**s, as **intelligent system**s, are able to detect known **target**s in adverse Radar conditions. These conditions are related with one of the most difficult **clutter** we can find, the coherent Weibull **clutter**. It is possible because **ANN**s trained in a supervised way can approximate the Neyman-Pearson (NP) detector (De la Mata-Moya, 2005, Vicen-Bueno, 2006, Vicen-Bueno, 2007), which is usually used in Radar systems design. This detector maximizes the probability of detection (Pd) maintaining the probability of false alarm (Pfa) lower than or equal to a given value (VanTrees, 1997). The detection of **target**s in presence of **clutter** is the main problem in Radar detection systems. Many **clutter** models have been proposed in the literature (Cheikh, 2004), although one of the most used models is the Weibull one (Farina, 1987a, DiFranco, 1980).

The research shown in (Farina, 1987b) set the optimum detector for **target** and **clutter** with arbitrary Probability Density Functions (PDFs). Due to the impossibility to obtain analytical expressions for the optimum detector, only suboptimum solutions were proposed. The Target Sequence Known A Priori (TSKAP) detector is one of them and is taken as reference for the experiments. Also, these solutions convey implementation problems, some of which make them non-realizable.

As mentioned above, one kind of **ANN**s, the **MultiLayer Perceptron** (**MLP**), is able to approximate the NP detector when it is trained in a supervised way to minimize the Mean Square Error (MSE) (Ruck, 1990, Jarabo, 2005). So, **MLP**s have been applied to the detection of known **target**s in different Radar environments (Gandhi, 1997, Andina, 1996).

INTELLIGENT RADAR DETECTORS BASED ON ARTIFICIAL NEURAL NETWORKS

This section starts with a discussion of the models selected for the **target**, **clutter** and **noise** signals. For these models, the optimum and suboptimum detectors are presented. These detectors will be taken as a reference for the experiments. After, it is presented the intelligent detector proposed in this work. This detector is based on **intelligent system**s like the **ANN**s, and a further analysis of its structure and parameters is made. Finally, several results are obtained for the detectors under study in order to analyze their performances.

Signal Models: Target, Clutter and Noise

The Radar is assumed to collect N pulses in a scan, so input vectors (z) are composed of N complex samples, which are presented to the detector. Under hypothesis H0 (target absent), z is composed of N samples of **clutter** and **noise**. Under hypothesis H1 (target present), a known **target** characterized by a fixed amplitude (A) and phase (θ) for each of the N pulses is summed up to the **clutter** and **noise** samples. Also, a Doppler frequency in the target model of $0,5 \cdot$ PRF is assumed, where PRF is the Pulse Repetition Frequency of the Radar system.

The **noise** is modelled as a coherent white Gaussian complex process of unity power, i.e., a power of ½ for the quadrature and phase components, respectively. The **clutter** is modelled as a coherent correlated sequence with Gaussian AutoCorrelation Function (ACF), whose complex samples have a modulus with a Weibull PDF:

$$p\left(|\mathbf{w}|\right) = ab^{-a} |\mathbf{w}|^{a-1} e^{-\left(\frac{|\mathbf{w}|}{b}\right)^a} \tag{1}$$

where |w| is the modulus of the coherent Weibull sequence and a and b are the skewness (shape) and scale parameters of a Weibull distribution, respectively.

The NxN autocorrelation matrix of the clutter is given by

$$\left(M_c\right)_{h,k} = P_c \rho_c^{|h-k|^2} e^{j\left(2\pi(h-k)\frac{f_c}{PRF}\right)} \tag{2}$$

where the indexes h and k varies from 1 to N, P_c is the clutter power, ρ_c is the one-lag correlation coefficient and f_c is the Doppler frequency of the clutter.

The relationship between the Weibull distribution parameters and P_c is

$$P_c = \frac{2b^2}{a} \Gamma\left(\frac{2}{a}\right) \tag{3}$$

where $\Gamma( )$ is the *Gamma function.*

The model used to generate coherent correlated Weibull sequences consists of two blocks in cascade: a correlator filter and a NonLinear MemoryLess Transformation (NLMLT) (Farina, 1987a). To obtain the desired sequence, a coherent white Gaussian sequence is correlated with the filter designed according to (2) and (3). The NLMLT block, according to (1), gives the desired Weibull distribution to the sequence. So, in that way, it is possible to obtained a coherent sequence with the desired correlation and PDF.

Taking into consideration that the complex **noise** samples are of unity variance (power), the following power relationships are considered for the study:

- **Signal to Noise Ratio**: SNR = $10\log_{10}(A^2)$
- **Clutter to Noise Ratio**: CNR = $10\log_{10}(P_c)$

Neyman-Pearson Detectors: Optimum and Suboptimum Detectors

The problem of optimum Radar detection of **target**s in **clutter** is explored in (Farina, 1987a) when both are time correlated and have arbitrary PDFs. The optimum detector scheme is built around two non-linear estimators of the disturbances in both hypotheses, which minimize the MSE. The study of Gaussian correlated **target**s detection in Gaussian correlated **clutter** plus **noise** is carried out, but for the cases where the hypothesis are non-Gaussian distributed, only suboptimum solutions are studied.

The proposed detectors basically consist of two channels. The upper channel is matched to the conditions that the sequence to be detected is the sum of the **target** plus **clutter** in presence of **noise** (hypothesis H1). While the lower one is matched to the detection of **clutter** in presence of **noise** (hypothesis H0).

For the detection problem considered in this paper, the suboptimum detection scheme (TSKAP) shown in figure 1 is taken. Considering that the **CNR** is very

Figure 1. Target sequence known a priori detector
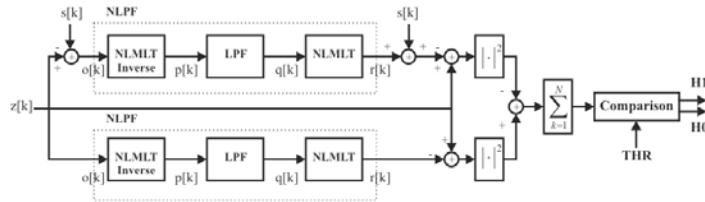
high (**CNR**>>1), the inverse of the NLMLT is assumed to transform the Weibull **clutter** in Gaussian, so the Linear Prediction Filter (LPF) is a N-1 order linear one. Then, the NLMLT transforms the filter output in a Weibull sequence. Besides being suboptimum, this scheme presents two important drawbacks:

1. The prediction filters have N-1 memory cells that must contain the suitable information to predict correct values for the N samples of each input pattern. So N+(N-1) pulses are necessary to decide if the target is present or not.
2. The target sequence must be subtracted from the input of the H1 channel.

There is no sense in subtracting the **target** component before deciding if this component is present or not. So, in practical cases, it makes this scheme non-realizable.

Intelligent Radar Detectors

In order to overcome the drawbacks of the scheme proposed in the previous section, a detector based on a **MLP** with log-sigmoid activation function in its hidden and output neurons with hard limit threshold after its output is proposed. Also, as **MLP**s have been probed to approximate the NP detector when minimizing the MSE (Jarabo, 2005), it can be expected that the MLP-based detector outperforms the suboptimum scheme proposed in (Farina, 1987a).

MLPs have been trained to minimize the MSE using two algorithms: the back-propagation (BP) with vary-

ing learning rate and momentum (Haykin, 1999) and the Levenberg-Marquardt (LM) with varying adaptive parameter (Bishop, 1995). While BP is based on the steepest descent method, the LM is based on the Newton method, which is designed specifically for minimizing the MSE. For **MLP**s which have up to few hundred of weights (W), the LM algorithm is more efficient than the BP one with variable learning rate or the conjugate gradient algorithms, being able to converge in many cases when the other two algorithms fail (Hagan, 1994). The LM algorithm uses the information (estimation of the WxW Hessian matrix) of the error surface in each iteration to find the minimum. It makes this algorithm faster than the previous ones.

Cross-validation is used with both training algorithms, where training and validation sets are synthetically generated. Moreover, a new set (test set) of patterns is generated to test the trained **MLP** for estimating the Pfa and Pd using Montecarlo simulation. All the patterns of the three sets are generated under the same conditions (**SNR**, **CNR** and a parameters of the Radar problem) in order to study the capabilities of the **MLP** plus hard limit thresholding working as a detector.

MLPs are initialized using the Nguyen-Widrow method (Nguyen, 1999) and, in all cases, the training process is repeated ten times to guarantee that the performance of all the **MLP**s is similar in average. Once all the **MLP**s are trained, the best **MLP** in terms of the estimated MSE with the validation set is selected, in order to avoid the problem of keeping in local minima at the end of the training.

The architecture of the **MLP** considered for the experiments is I/H/O, where I is the number of **MLP**

inputs, H is the number of hidden neurons in its hidden layer and O is the number of **MLP** outputs. As the **MLP**s work with real arithmetic, if the input vector (z) is composed of N complex samples, the **MLP** will have 2N inputs (N in phase and N in quadrature components). The number of **MLP** independent elements (weights) to solve the problem is W=(I+1)·H+(H+1)·O, including the bias of each neuron.

Results

The performance of the detectors exposed in the previous sections is shown in terms of the Receiver Operating Characteristics (ROC) curves. They give the estimated Pd for a desired Pfa, which values are obtained varying the output threshold of the detector. The experiments presented are made for an integration of two pulses (N=2). So, in order to test correctly the TSKAP detector, observation vectors (also called patterns during the text) of length 3 (N+(N-1)) complex samples are generated, due to memory requirements of the TSKAP detector (N-1 pulses).

The a priori probabilities of H0 and H1 hypothesis are supposed to be the same. Three sets of patterns are generated for each experiment: train, validation and test sets. The first and the second ones have $5 \cdot 10^3$ patterns, respectively. The third one has $2.5 \cdot 10^6$ patterns, so the error in the estimation of the Pfa and the Pd is lower than 10% of the estimated values in the worst case (Pfa=10^{-4}). The patterns of all the sets are synthetically generated under the same conditions. These conditions involve typical values (Farina, 1987a, DiFranco, 1980, Farina, 1987b) for the **SNR** (20 dB), the **CNR** (30 dB) and the a (a=1.2) parameter of the Weibull-distributed **clutter**.

The **MLP** architecture used to generate the MLP-based detector is 6/H/1. The number of **MLP** outputs (O=1) is established by the problem (binary detection). The number of hidden neurons (H) is studied in this work. And the number of **MLP** inputs (I=6) is established according to the next criterion. A total of 6 inputs (2(N+(N-1))) are selected when the MLP-based detector wants to be compared with the TSKAP detector in the same conditions, i.e., when both detectors have the same available information (3 pulses for an integration of N=2 pulses). Because of the TSKAP detector memory requirements, this case is considered.

Figure 2 shows the results of a study when 3 pulses are used to take the final decision by the MLP-based

detector according to the criterion exposed above. The study shows the influence of the training algorithm and the **MLP** size, i.e., the number of independent elements (W weights) that has the **ANN** to solve the problem. For the case of study, two important aspects have to be noted. The first one is related with the training algorithm. As can be observed, the performance achieved with a low size **MLP** (6/05/1) is very similar for both training algorithms (LM and BP). But when the **MLP** size is greater, for instance, 6/10/1, the performance achieved with the LM algorithm is better than the performance achieved with the BP one. It is due to the LM algorithm is more efficient than the BP one finding the minimum of the error surface. Moreover, the **MLP** training with LM is faster than the training with BP, because the number of training epochs can be reduced in an order of magnitude. The second aspect is related with the **MLP** size. As can be observed, no performance improvement is achieved when 20 or more hidden neurons are used comparing both algorithms as occurred with 10 hidden neurons. Moreover, from 20 (W=121 weights) to 30 (W=181 weights) hidden neurons, the performance tends to a maximum value (independently of the training algorithm used), i.e., almost no performance improvement is achieved with more weights. So, an MLP-based detector with 20 hidden neurons achieves an appropriate performance with low complexity.

A comparison between the performances achieved with the TSKAP detector and the MLP-based detector of size 6/20/1 trained with BP and LM algorithms is shown in figure 3. Two differences can be observed. The first one is that the MLP-based detector performance is practically independent of the training algorithm, comparing their results with the ones obtained for the TSKAP detector. And the second one is that the 6/20/1 MLP-based detector is always better than the TSKAP detector when they are compared in the same conditions of availability of information, i.e., with the availability of 3 (N+(N-1)) pulses to decide. Under these conditions and comparing figures 2 and 3, it can be observed that a 6/05/1 MLP-based detector is enough to overcome the TSKAP one.

The appreciated differences between the TSKAP and MLP-based detectors appear because the first one is a suboptimum detector and the second one approximates the optimum one, but it will be always worse than the optimum detector. It can not be demonstrated because an analytical expression for the optimum detector

Figure 2. MLP-based detector performances for different structure sizes (6/H/1) and different training algorithms: (a) BP and (b) LM
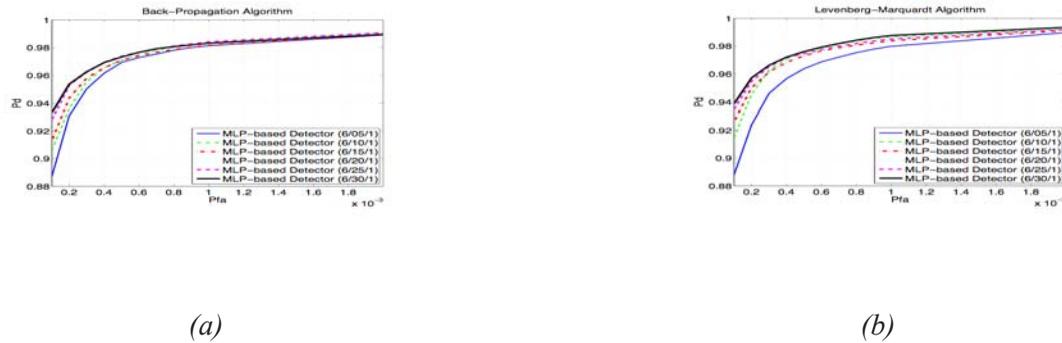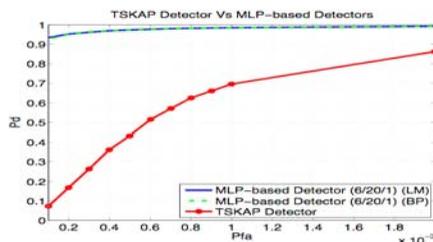

(a)

(b)

Figure 3. TSKAP and MLP-based detectors performances for MLP size 6/20/1 trained with BP and LM algorithms

can not be obtained detecting **target**s in presence of Weibull-distributed **clutter**.

FUTURE TRENDS

Two different future trends can be mentioned. The first one is related with **ANN**s and the second one is related with the research in Radar detectors. In the first trend, it is possible to emphasize the research in areas like ensembles of **ANN**s, committee machines based on **ANN**s and others way to combine the intelligence of different **ANN**s like the **MLP**s, the Radial Basis Functions and others. Moreover, new trends try to find different ways to train **ANN**s. In the second trend, several researchers are trying to find different ways to create radar detectors in order to improve their performances. Moreover, several solutions are proposed, but they depend on the Radar environment considered. So, detectors based on signal processing tools seem to be the most appropriated, but the intelligent detector exposed here is a new way of working, which can brings good solutions to these problems. This is possible because of the intelligence of the **ANN**s to adapt to almost any kind of Radar conditions and problems.

CONCLUSION

After the developed study, several conclusions can be set. The LM training algorithm achieves better MLP-based detectors than the BP one. No performance improvement is obtained for training MLPs with LM or BP algorithms when their sizes are greater than 6/20/1. But, the great advantage of the LM one against the BP one is its fastest training for low size MLPs (a

few hundred of weights), i.e., the MLPs considered in this study. Finally, the MLP-based detector works better than the TSKAP one in cases of working with the same available information (N+(N-1)=3), because the memory requirements of the TSKAP one. In those cases, low complexity MLP-based detectors can be obtained because a 6/05/1 MLP has enough intelligence to obtain better performance than the TSKAP one.

REFERENCES

Andina, D., & Sanz-Gonzalez, J.L. (1996). Comparison of a Neural Network Detector Vs Neyman-Pearson Optimal Detector. *Proc. of ICASSP-96*. 3573-3576.

Bishop, C.M. (1995). *Neural networks for pattern recognition.* Oxford University Press Inc.

De la Mata-Moya, D., Jarabo-Amores, P., Rosa-Zurera, M., López-Ferreras, F., & Vicen-Bueno, R. (2005). Approximating the Neyman-Pearson Detector for Swerling I Targets with Low Complexity Neural Networks. *Lecture Notes in Computer Science*. (3697), 917-922.

Cheikh, K., & Faozi S. (2004). Application of Neural Networks to Radar Signal Detection in K-distributed Clutter. *First Int. Symp. on Control, Communications and Signal Processing Workshop Proc.* 633-637.

DiFranco, J.V., & Rubin, W.L. (1980). *Radar Detection.* Artech House.

Farina, A.,Russo, A., Scannapieco, F., & Barbarossa, S. (1987a). Theory of Radar Detection in Coherent Weibull Clutter. *In: Farina, A. (eds.): Optimised Radar Processors. IEE Radar, Sonar, Navigation and Avionics, Series 1. Peter Peregrinus Ltd.* 100-116.

Farina, A., Russo, A., & Scannapieco, F. (1987b). Radar Detection in Coherent Weibull Clutter. *IEEE Trans. on Acoustics, Speech and Signal Processing.* ASSP-35 (6), 893-895.

Gandhi, P.P., & Ramamurti, V. (1997). Neural Networks for Signal Detection in Non-Gaussian Noise. *IEEE Trans. on Signal Processing*. (45) 11, 2846-2851.

Hagan. M.T., & Menhaj, M.B. (1994). Training Feedforward Networks with Marquardt Algorithm. *IEEE Trans. on Neural Networks*. (5) 6, 989-993.

Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation (Second Edition)*. Prentice-Hall.

Jarabo-Amores, P., Rosa-Zurera, M., Gil-Pita, R., & López-Ferreras, F. (2005). Sufficient Condition for an Adaptive System to Aproximate the Neyman-Pearson Detector. *Proc. IEEE Workshop on Statistical Signal Processing*. 295-300.

Nguyen, D., & Widrow, B. (1999). Improving the Learning Speed of 2-layer Neural Networks by Choosing Initial Values of the Adaptive Weights. *Proc. of the Int. Joint Conf. on Neural Networks.* 21-26.

Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., & Suter, B.W. (1990). The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function. *IEEE Trans. on Neural Networks.* (1) 11, 296-298.

Van Trees, H.L. (1997). *Detection, Estimation and Modulation Theory. Part I.* John Wiley and Sons.

Vicen-Bueno, R., Jarabo-Amores, M. P., Rosa-Zurera, M., Gil-Pita, R., & Mata-Moya, D. (2007). Performance Analysis of MLP-Based Radar Detectors in Weibull-Distributed Clutter with Respect to Target Doppler Frequency. *Lecture Notes in Computer Science*. (4669), 690-698.

Vicen-Bueno, R., Rosa-Zurera, M., Jarabo-Amores, P., & Gil-Pita, R. (2006). NN-Based Detector for Known Targets in Coherent Weibull Clutter. *Lecture Notes in Computer Science*. (4224), 522-529.

KEY TERMS

Artificial Neural Networks (ANNs): A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Backpropagation Algorithm: Learning algorithm of ANNs, based on minimising the error obtained from the comparison between the ANN outputs after the application of a set of network inputs and the desired

outputs. The update of the weights is done according to the gradient of the error function evaluated in the point of the input space that indicates the input to the ANN.

Knowledge Extraction: Explicitation of the internal knowledge of a system or set of data in a way that is easily interpretable by the user.

Intelligence: It is a property of mind that encompasses many related abilities, such as the capacities to reason, plan, solve problems, think abstractly, comprehend ideas and language, and learn.

Levenberg-Marquardt Algorithm: Similar to the Backpropagation algorithm, but with the difference that the error is estimated according to the Hessian Matrix. This matrix gives information of several directions where to go in order to find the minimum of the error function, instead of the local minimum one that gives the backpropagation algorithm.

Probability Density Function: The statistical function that shows how the density of possible observations in a population is distributed.

Radar: It is the acronym of Radio Detection and Ranging. In few words, a Radar emits an electromagnetic wave that is reflected by the target and others objects present in its observation space. Finally, the Radar receives these reflected waves (echoes) to analyze them in order to decide whether a target is present or not.

Intelligent Software Agents Analysis in E-Commerce I

Xin Luo
The University of New Mexico, USA

Somasheker Akkaladevi
Virginia State University, USA

INTRODUCTION

Equipped with sophisticated information technology infrastructures, the information world is becoming more expansive and widely interconnected. Internet usage is expanding throughout the web-linked globe, which stimulates people's need for desired information in a timely and convenient manner. Electronic commerce activities, powered by Internet growth, are increasing continuously. It is estimated that online retail will reach nearly $230 billion and account for 10% of total U.S. retail sales by 2008 (Johnson et al. 2003). In addition, e-commerce entailing business-to-business (B2B), business-to-customer (B2C) and customer-to-customer (C2C) transactions is spawning new markets such as mobile commerce.

By increasing the degree and sophistication of the automation, commerce becomes much more dynamic, personalized, and context sensitive for both buyers and sellers. Software agents were first used several years ago to filter information, match people with similar interests, and automate repetitive behavior (Maes et al. 1999). In recent years, agents have been applied to the arena of e-commerce, triggering a revolutionary change in the way we conduct online transactions in B2B, B2C, and C2C. Researchers argue that the potential of the Internet for transforming commerce is largely unrealized (Begin et al. 2002; Maes et al. 1999). Further, He and Jennings noted that a new model of software agent is needed to achieve the degree of automation and move to second generation e-commerce1 applications (He et al. 2003). This is due to the predicament that electronic purchases are still largely unautomated. Maes et al. (1999) also addressed that, even though information is more easily accessible and orders and payments are dealt with electronically, humans are still in the loop in all stages of the buying process, which inevitably increase the transaction costs. Undoubtedly,

a human buyer is still responsible for collecting and interpreting information on merchants and products, making decisions about merchants and products, and ultimately entering purchase and payment information. Additionally, Jennings et al. (1998) confirmed that commerce is almost entirely driven by human interactions and further argued that there is no reason why some commerce cannot be automated.

This unautomated loop requires a lot of time and energy and results in inefficiency and high cost for both buyers and sellers. To automate time-consuming tasks, intelligent software agent (ISA) technology can play an important role in online transaction and negotiation due to its capability of delivering unprecedented levels of autonomy, customization, and general sophistication in the way e-commerce is conducted (Sierra et al. 2003). Systems containing ISAs have been developed to automate the complex process of negotiating a deal between a buyer and a seller. An increasing number of e-commerce agent systems are being developed to support online transactions that have a number of variables to consider and to aim for a win-win result for sellers and buyers.

In today's e-commerce arena, systems equipped with ISAs may allow buyers and sellers to find the best deal taking into account the relative importance of each factor. Advanced systems of e-commerce that embody ISA technologies are able to perform a number of queries and to process phenomenal volumes of information. ISAs reduce transaction costs by collecting information about services and commodities from a lot of firms and presenting only those results with high relevance to the user. ISA technologies help businesses automate information transaction activity, largely eliminate human intervention in negotiation, lower transaction and information search cost, and further cultivate competitive advantage for companies. Therefore, ISAs can free people to concentrate on the

issues requiring true human intelligence and intervention. Implementing the personalized, social, continuously running, and semi-autonomous ISA technologies in business information systems, the online business can become more user-friendly, semi-intelligent, and human-like (Pivk 2003).

LITERATURE REVIEW

A number of scholars have defined the term *intelligent software agent*. Bradshaw (1997) proposed that one person's intelligent agent is another person's smart object. Jennings and Wooldridge (1995) defined agents as a computer system situated in some environment that is capable of autonomous action in this environment to meets its design objective. Shoham (1997) further described an ISA as a software entity which functions continuously and autonomously in a particular environment, often inhabited by other agents and processes. In general, an ISA is *a software agent that uses Artificial Intelligence (AI) in the pursuit of the goals of its clients* (Croft 2002). It can perform tasks independently on behalf of a user in a network and help users with information overload. It is different from current programs in terms of being proactive, adaptive, and personalized (Guttman et al. 1998b). Also, it can actively initiate actions for its users according to the configurations set by the users; it can read and understand user's preferences and habits to better cater to user's needs; it can provide the users with relevant information according to the pattern it adapts from the users.

ISA is a cutting-edge technology in computational sciences and holds considerable potential to develop new avenues in information and communication technology (Shih et al. 2003). It is used to perform multi-task operations in decentralized information systems, such as the Internet, to conduct complicated and wide-scale search and retrieval activities, and assist in shopping decision-making and product information search (Cowan et al. 2002). ISA's ability of performing continuously and autonomously stems from human desire in that an agent is capable of operating certain activities in a flexile and intelligent manner responsive to changes in the environment without constant human supervision. Over a long period of time, an agent is capable of adapting from its previous experience and would be able to inhabit an environment with other agents to communicate and cooperate with them to achieve tasks for human.

Intelligent Agent Taxonomy and Typology

Franklin and Grasser (1996) proposed a general taxonomy of agent (see Figure 1).

This taxonomy is based on the fact that ISA technologies are implemented in a variety of areas, including biotechnology, economic simulation and data-mining, as well as in hostile applications (malicious codes), machine learning and cryptography algorithms. In addition, Nwana (1996b) proposed the agent typology (see Figure 2) in which four types of agents can be categorized: collaborative agents, collaborative learning agents, interface agents and smart agents. These four agents have different congruence amid learning, autonomy, and cooperation and therefore tend to address different sides of this topology in terms of the functionality.

According to Nwana (1996b), collaborative agents emphasize more autonomy and cooperation than learning. They collaborate with other agents in multi-agent environments and may have to negotiate with other agents in order to reach mutually acceptable agreements for users. Unlike collaborative agents, interface agents emphasize more autonomy and learning. They support and provide proactive assistance. They can observe user's actions in the interface and suggest better ways for completing a task for the user. Also, interface agents' cooperation with other agents is typically limited to asking for advice (Ndumu et al. 1997).

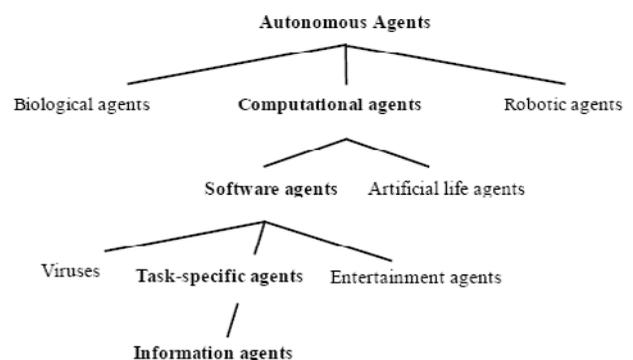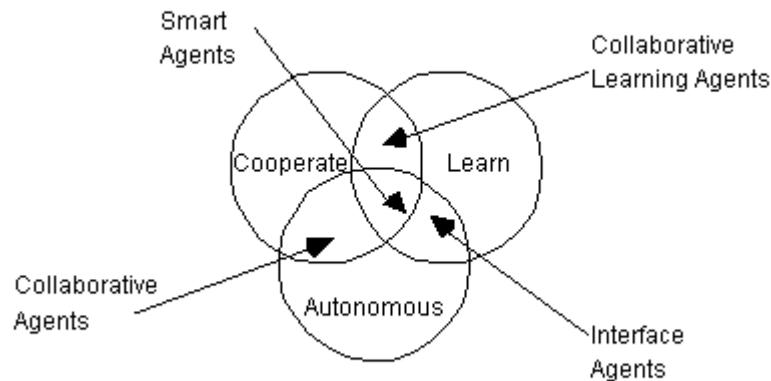*Figure 1. Franklin and Grasser's agent taxonomy (Source: Franklin & Grasser. 1996)*

Figure 2. A Part View of Agent Typology Source: Nwana (1996b)

The benefits of interface agents include reducing user's efforts in repetitive work and adapting to their user's preferences and habits. Smart agents are agents that are intelligent, adaptive, and computational (Carley 1998). They are advanced intelligent agents summing up the best capabilities and properties of all presented categories.

This proposed typology highlights the key contexts in which the agent is used in AI literature. Yet Nwana (1996b) argued that agents ideally should do all three equally well, but this is the aspiration rather than the reality. Furthermore, according to Nwana (1996b) and Jennings and Wooldridge (1998), five more agent types could be derived based on the typology, from a panoramic perspective (see Figure 3).

In this proposed typology, mobile agents are autonomous and cooperative software processes capable of roaming wide area networks, interacting with foreign hosts, performing tasks on behalf of their owners (Houmb 2002). Information agents can help us manage the explosive growth of information we are experiencing. They perform the role of managing, manipulating, or collating information from many distributes sources (Nwana 1996b). Reactive agents choose actions by using the current world state as an index into a table of actions, where the indexing function's purpose is to map known situations to appropriate actions. These types of agents are sufficient for limited environments where every possible situation can be mapped to an action or set of actions (Chelberg 2003). Hybrid agents adopt strength of both the reactive and deliberative paradigms. They aim to have the quick response time of reactive agents for well known situations, yet also have the ability to generate new plans for unforeseen situations (Chelberg 2003). Heterogeneous agents systems refer to an integrated set-up of at least two or more agents, which belong to two or more different agent classes (Nwana 1996b).

CONCLUSION AND FUTURE WORK

This paper explores how ISAs can automate and add value to e-commerce transactions and negotiations. By

Figure 3. A panoramic overview of the different agent types (Source: Jennings & Wooldridge, 1998)

leveraging ISA-based e-commerce systems, companies can more efficiently make decisions because they have more accurate information and identify consumers' tastes and habits. Opportunities and limitations for ISA development are also discussed. Future technologies of ISAs will be able to evaluate basic characteristics of online transactions in terms of price and product description as well as other properties, such as warranty, method of payment, and after-sales service. Also, they would better manage ambiguous content, personalized preferences, complex goals, changing environments, and disconnected parties (Guttman et al. 1998a). Additionally, for the future trend of ISA technology deployment, Nwana (1996a) describes that *"Agents are here to stay, not least because of their diversity, their wide range of applicability and the broad spectrum of companies investing in them. As we move further and further into the information age, any information-based organization which does not invest in agent technology may be committing commercial hara-kiri."*

REFERENCES

Begin, L., and Boisvert, H. "Enhancing the value proposition Via the internet," *International Conference on Electronic Commerce Research (ICECR-5),* 2002.

Bradshaw, J.M. "Software Agents," *online: http://agents.umbc.edu/introduction/01-Bradshaw.pdf*) 1997.

Carley, K.M. "Smart Agents and Organizations of the Future," *online: http://www.hss.cmu.edu/departments/sds/faculty/carley/publications/ORGTHEO36.pdf*) 1998.

Chelberg, D. "Reactive Agents," *online: http://zen.ece.ohiou.edu/~robocup/papers/HTML/AAAI/node3.html*), 03-05 2003.

Cowan, R., and Harison, E. "Intellectual Property Rights in Intelligent-Agent Technologies: Facilitators, Impediments and Conflicts," *online: http://www.itas.fzk.de/e-society/preprints/ecommerce/CowanHarison.pdf*) 2002.

Croft, D.W. "Intelligent Software Agents: Definitions and Applications," *online: http://www.alumni.caltech.edu/~croft/research/agent/definition/*) 2002.

Franklin, S., and Graesser, A. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents," *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag,* 1996.

Guttman, R., Moukas, A., and Maes, P. "Agent-mediated Electronic Commerce: A Survey," *Knowledge Engineering Review* (13:3), June 1998a.

Guttman, R., Moukas, A., and Maes, P. "Agents as Mediators in Electronic Commerce," *International Journal of Electronic Markets* (8:1), February 1998b, pp 22-27.

He, M., Jennings, N.R., and Leung, H.-F. "On Agent-Mediated Electronic Commerce," *IEEE Transactions on Knowledge and Data Engineering* (15:4), July/August 2003.

Houmb, S.H. "Software Agent: An Overview," *online: http://www.idi.ntnu.no/emner/dif8914/ppt-2002/sw-agent_dif8914_2002.ppt*) 2002.

Jennings, N.R., and Wooldridge, M. "Applications of Intelligent Agents," *in Agent Technology: Foundations, Applications, and Markets,* 1998, pp 3-28.

Johnson, C., Delhagen, K., and Yuen, E.H. "US eCommerce Overview: 2003 To 2008," *Online: http://www.forrester.com/ER/Research/Brief/Excerpt/0,1317,16875,00.html*), July 25 2003.

Maes, P., Guttnab, R.H., and Moukas, A.G. "Agents That Buy and Sell. (software agents for electronic commerce)(Technology Information)," *Communications of the ACM* (42:3) 1999, p 81.

Ndumu, D., and Nwana, H. "Research and Development Challenges for Agent-Based Systems," *IEE Proceedings on Software Engineering* (144:01), January 1997.

Nwana, H.S. "Software Agents: An Overview," *online: http://agents.umbc.edu/introduction/ao/*) 1996b.

Pivk, A. "Intelligent Agents in E-Commerce," *online: http://ai.ijs.si/Sandi/IntelligentAgentRepository.html*) 2003.

Shih, T.K., Chiu, C.-F., and Hsu, H.-h. "An Agent-Based Multi-Issue Negotiation System in E-commerce," *Journal of Electronic Commerce in Organizations* (1:1), Jan-March 2003, pp 1-16.

Sierra, C., Wooldridge, M., Sadeh, N., Conte, R., Klusch, M., and Treur, J. "Agent Research and Development in Euroope," *online: http://www.unicom.co.uk/3in/IS-SUE4/4.Asp*) 2003.

KEY TERMS

Agent: A computer system situated in some environment that is capable of autonomous action in this environment to meets its design objective.

Business-to-Business E-Commerce: Electronic transaction of goods or services between businesses as opposed to that between businesses and other groups.

Business-to-Customer E-Commerce: Electronic or online activities of commercial organizations serving the end consumer with products and/or services. It is usually applied exclusively to e-commerce.

Customer-to-Customer E-Commerce: Online transactions involving the electronically-facilitated transactions between consumers through some third party.

Electronic Commerce (E-Commerce): Consists of the buying and selling of products or services over electronic systems such as the Internet and other computer networks. A wide variety of commerce is conducted in this way, including electronic funds transfer, supply chain management, e-marketing, online transaction processing, and automated data collection systems.

Intelligent Software Agent: A software agent that uses Artificial Intelligence (AI) in the pursuit of the goals of its clients.

Ubiquitous Commerce (U-Commerce): The ultimate form of e-commerce and m-commerce in an 'anytime, anywhere' fashion. It involves the use of ubiquitous networks to support personalized and un-interrupted communications and transactions at a level of value that far exceeds traditional commerce.

Intelligent Software Agents Analysis in E–Commerce II

Xin Luo
The University of New Mexico, USA

Somasheker Akkaladevi
Virginia State University, USA

ISA OPPORTUNITIES AND LIMITATIONS IN E-COMMERCE

Cowan et al. (2002) argued that the human cognitive ability to search for information and to evaluate their usefulness is extremely limited in comparison to those of computers. In detail, it's cumbersome and time-consuming for a person to search for information from limited resources and to evaluate the information's usefulness. They further indicated that while people are able to perform several queries in parallel and are good at drawing parallels and analogies between pieces of information, advanced systems that embody ISA architecture are far more effective in terms of calculation power and parallel processing abilities, particularly in the quantities of material they can process (Cowan et al. 2002). According to Bradshaw (1997), information complexity will continue to increase dramatically in the coming decades. He further contended that the dynamic and distributed nature of both data and applications require that software not merely respond to requests for information but intelligently anticipate, adapt, and actively seek ways to support users.

E-commerce applications based on agent-oriented e-commerce systems have great potential. Agents can be designed using the latest web-based technologies, such as Java, XML, and HTTP, and can dynamically discover and compose E-services and mediate interactions to handle routine tasks, monitor activities, set up contracts, execute business processes, and find the best services (Shih et al., 2003). The main advantages of using these technologies are their simplicity of usage, ubiquitous nature, and their heterogeneity and platform independence (Begin and Boisvert, 2002). XML will likely become the standard language for agent-oriented E-commerce interactions to encode exchanged messages, documents, invoices, orders, service descriptions, and other information. HTTP, the dominant WWW protocol, can be used to provide many services, such as robust and scalable web servers, firewall access, and levels of security for these E-commerce applications.

Agents can be made to work individually, as well as in a collaborative manner to perform more complex tasks (Franklin and Graesser, 1996). For example, to purchase a product on the Internet, a group of agents can exchange messages in a conversation to find the best deal, can bid in an auction for the product, can arrange financing, can select a shipper, and can also track the order. Multi-agent systems (groups of agents collaborating to achieve some purpose) are critical for large-scale e-commerce applications, especially B2B interactions such as service provisioning, supply chain, negotiation, and fulfillment, etc. The grouping of agents can be static or dynamic depending on the specific need (Guttman et al., 1998b). A perfect coordination should be established for the interactions between the agents to achieve a higher-level task, such as requesting, offering and accepting a contract for some services (Guttman et al., 1998a).

There are several agent toolkits publicly available which can be used to satisfy the customer requirements and ideally they need to adhere to standards which define multi-party agent interoperability. For example, fuzzy logic based intelligent negotiation agents can be used to interact autonomously and consequently, and save human labor in negotiations. The aim of modeling a negotiation agent is to reach mutual agreement efficiently and intelligently. The negotiation agent should be able to negotiate with other such agents over various sets of issues, and on behalf of the real-world parties they represent, i.e. they should be able to handle multi-issue negotiations at any given time.

The boom in e-commerce has now created the need for ISAs that can handle complicated online transactions and negotiations for both sellers and buyers. In

general, buyers want to find sellers that have desired products and services. And they want to find product information and gain expert advice before and after the purchase from sellers, which, in turn, want to find buyers and provide expert advice about their product or service as well as customer service and support. Therefore, there is an opportunity that both buyers and sellers can automate handling this potential transaction by adopting ISA technology. The use of ISAs will be essential to handling many tasks of creating, maintaining, and delivering information on the Web. By implementing ISA technology in e-commerce, agents can shop around for their users; they can communicate with other agents for product specifications, such as price, feature, quantity, and service package, and make a comparison according to user's objective and requirement and return with recommendations of purchases, which can meet those specifications; they can also act for sellers by providing product or service sales advice, and help troubleshoot customer problems by automatically offering solutions or suggestions; they can automatically pay bills and keep track of the payment.

Looking at ISA development from an international stand point, the nature of Internet in developed countries, such as USA, Canada, West Europe, Japan, and Australia, etc. and the consequent evolution of e-commerce as the new model provide exciting opportunities and challenges for ISA-based developments. Opportunities include wider market reach in a timely manner, higher earnings, broader spectrum of target and potential customers, and collaboration among vendors. This ISA-powered e-commerce arena would be different than our traditional commerce, because the traditional form of competition can give way to collaborative efforts across industries for adding value to business processes. This means that agents of different vendors can establish a cooperative relationship to communicate with each other via XML language in order to set up and complete transactions online.

Technically, for instance, if an information agent found that the vendor is in need of more airplane tickets, it would notify a collaborative agent to search for relevant information regarding the ticket in terms of availability, price, and quantity etc. from other sources over the Internet. In this case, the collaborative agent would work with mobile agents and negotiate with other agents working for different vendors and obtain ticket information for its user. It would be able to provide

the user with the result of the search, and, if needed, purchase the tickets for the user if certain requirements can be met. In the meantime, interface agents can monitor the user's reaction and decision behavior, and would provide the user with informational assistance in terms of advice, recommendation, and suggestion for any related and similar transactions.

On the other hand, however, this kind of intelligent electronic communication and transaction is relatively inapplicable in traditional commerce where different competitive vendors are not willing to share information with each other (Maes et al., 1999). The level of willingness in ISA-based e-commerce is, however, somewhat limited due to sociological and ethical factors, which will be discussed later in this paper. In addition, designing and implementing ISA technology is a costly predicament preventing companies from adopting this emerging tool. Companies need to invest a lot of money to get the ISA engine started. Notwithstanding the exciting theoretical benefits discussed above, many companies are still not sure about how much ISA technology can benefit themselves in terms of revenue, ROI, and business influence in the market where other players are yet to adopt this technology to cooperate with each other. Particularly, medium or small size companies are reluctant to embark on this arena mainly due to the factor of cost.

Additionally, lack of consistent architectures in terms of standards and laws also obstructs the further development of ISA technology (He et al., 2003). In detail, IT industry has not yet finalized the ISA standards, as there are a number of proprietary standards set by various companies. This causes a confusion problem for ISAs to freely communicate with each other. Also, related to standards, relevant laws have not surfaced to regulate how ISAs can legally cooperate with each other and represent their human users in the cyber world.

Additionally, ISA development and deployment is not a global perspective (Jennings et al. 1998). Despite the fact that ISA technology is an ad-hoc topic in developed countries, developing countries are not fully aware of the benefits of ISA and therefore have not deployed ISA-based systems on the Web because their e-commerce development levels and skills are not as sophisticated or advanced as those of the developed countries. This intra-national limitation among developed and developing countries unfortunately hinders

agents from freely communicating with each other over the globally connected Internet.

SOCIOLOGICAL AND ETHICAL CHALLENGES

In the preceding sections of this paper, the technical issues involved in agent development have been addressed. However, in addition to these issues, there are also a range of social and cyber-ethical problems, such as trust and delegation, privacy, responsibility, and legal issues, which will become increasingly important in the field of agent technology (Bradshaw 1997; Jennings et al. 1998; Nwana 1996b).

- **Trust and delegation:** For users who want to depend on ISA technology to obtain desired information, they must trust agents which autonomously delegate for users to do the job. It would take time for users to get used to their agents and gain confidence in the agents that work for them. And users have to make a balance between agents continually seeking guidance and never seeking guidance. Users might need to set proper limitations for their agents, otherwise agents might surpass their authorities.
- **Privacy:** In the explosive information society, security is becoming more and more important. Therefore, users must make sure that their agents always maintain their privacy in the course of transactions. Electronic agent security policies may be needed to encounter this potential threat.
- **Responsibility:** Users need to seriously consider how much responsibility the agents need to carry regarding the transaction pitfall. To some extent, agents are rendered responsibility to get the desired product/service for their users. If the users are not satisfied with the transaction result, they may need to redesign or reprogram the agent rather than directly blame the fault on electronic agents.
- **Legal issues:** In addition to responsibility, users should also think about any potential legal issues triggered by their agents, which, for instance, offer inappropriate advice to other agents resulting in liabilities to other people. This would be very challenging to the ISA technology development, and the scenario would be complicated since the current law does not specify which party (the

company who wrote the agent, the company who customized and used the agent, or both) should be responsible for the legal issues.
- **Cyber-ethical issues:** Eichmann (1994) and Etzioni & Weld (1994) proposed the following etiquettes for ISAs which gather information on the Web.
  - Agents must identify themselves;
  - They must moderate the pace and frequency of their requests to some server;
  - They must limit their searches to appropriate servers;
  - They must share information with others;
  - They must respect the authority placed on them by server operators;
  - Their services must be accurate and up-to-date;
  - Safety: they should not destructively alter the world;
  - Tidiness: they should leave the world as they found it;
  - Thrift: they should limit their consumption of scarce resources;
  - Vigilance: they should not allow client actions with unanticipated results.

CONCLUSION AND FUTURE WORK

ISA technology has to confront the increasing complexity of modern information environments. Research and development of ISAs on the Internet is crucial for the development of next generation in open information environments. Sociological and cyber-ethical issues need to be considered for the next generation of agents in e-commerce system, which will explore new types of transactions in the form of dynamic relationships among previously unknown parties (Guttman et al. 1998b). According to Nwana (1996a), the ultimate ISA's success will be the acceptance and mass usage by users, once issues such as privacy, trust, legal, and responsibility are addressed and considered when users design and implement ISA technologies in e-commerce and emerging commerce, such as mobile commerce (M-commerce) and Ubiquitous commerce (U-commerce). It is expected that future research can further explore how ISAs are leveraged in these two newly emerged avenues.

REFERENCES

Begin, L., and Boisvert, H. "Enhancing the value proposition Via the internet," *International Conference on Electronic Commerce Research (ICECR-5),* 2002.

Bradshaw, J.M. "Software Agents," *online: http://agents.umbc.edu/introduction/01-Bradshaw.pdf*) 1997.

Cowan, R., and Harison, E. "Intellectual Property Rights in Intelligent-Agent Technologies: Facilitators, Impediments and Conflicts," *online: http://www.itas.fzk.de/e-society/preprints/ecommerce/CowanHarison.pdf*) 2002.

Eichmann, D. "Ethical Web Agents," *Second International World-Wide Web Conference: Mosaic and the Web,* October 18-20 1994, pp 3-13.

Franklin, S., and Graesser, A. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents," *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag,* 1996.

Etzioni, O., and Weld, D. "A Softbot-Based Interface to the Internet," *Communications of the ACM,* July 1994, pp 72-76.

Guttman, R., Moukas, A., and Maes, P. "Agent-mediated Electronic Commerce: A Survey," *Knowledge Engineering Review* (13:3), June 1998a.

Guttman, R., Moukas, A., and Maes, P. "Agents as Mediators in Electronic Commerce," *International Journal of Electronic Markets* (8:1), February 1998b, pp 22-27.

He, M., Jennings, N.R., and Leung, H.-F. "On Agent-Mediated Electronic Commerce," *IEEE Transactions on Knowledge and Data Engineering* (15:4), July/August 2003.

Jennings, N.R., and Wooldridge, M. "Applications of Intelligent Agents," *in Agent Technology: Foundations, Applications, and Markets,* 1998, pp 3-28.

Maes, P., Guttnab, R.H., and Moukas, A.G. "Agents That Buy and Sell. (software agents for electronic commerce)(Technology Information)," *Communications of the ACM* (42:3) 1999, p 81.

Ndumu, D., and Nwana, H. "Research and Development Challenges for Agent-Based Systems," *IEE Proceedings on Software Engineering* (144:01), January 1997.

Nwana, H.S. "Intelligent Software Agents on the Internet: an inventory of currently offered functionality in the information society & a prediction of (near-) future developments," *online: http://www.hermans.org/agents/index.html*), July 1996a.

Nwana, H.S. "Software Agents: An Overview," *online: http://agents.umbc.edu/introduction/ao/*) 1996b.

Shih, T.K., Chiu, C.-F., and Hsu, H.-h. "An Agent-Based Multi-Issue Negotiation System in E-commerce," *Journal of Electronic Commerce in Organizations* (1:1), Jan-March 2003, pp 1-16.

KEY TERMS

Agent: A computer system situated in some environment that is capable of autonomous action in this environment to meets its design objective.

Business-to-Business E-Commerce: Electronic transaction of goods or services between businesses as opposed to that between businesses and other groups.

Business-to-Customer E-Commerce: Electronic or online activities of commercial organizations serving the end consumer with products and/or services. It is usually applied exclusively to e-commerce.

Customer-to-Customer E-Commerce: Online transactions involving the electronically-facilitated transactions between consumers through some third party.

Electronic Commerce (E-Commerce): Consists of the buying and selling of products or services over electronic systems such as the Internet and other computer networks. A wide variety of commerce is conducted in this way, including electronic funds transfer, supply chain management, e-marketing, online transaction processing, and automated data collection systems.

Intelligent Software Agent: A software agent that uses Artificial Intelligence (AI) in the pursuit of the goals of its clients.

Ubiquitous Commerce (U-Commerce): The ultimate form of e-commerce and m-commerce in an 'anytime, anywhere' fashion. It involves the use of ubiquitous networks to support personalized and uninterrupted communications and transactions at a level of value that far exceeds traditional commerce.

Intelligent Software Agents with Applications in Focus

Mario Janković-Romano
University of Belgrade, Serbia

Milan Stanković
University of Belgrade, Serbia

Uroš Krčadinac
University of Belgrade, Serbia

INTRODUCTION

Most people are familiar with the concept of agents in real life. There are stock-market agents, sports agents, real-estate agents, etc. Agents are used to filter and present information to consumers. Likewise, during the last couple of decades, people have developed software agents, that have the similar role. They behave intelligently, run on computers, and are autonomous, but are not human beings.

Basically, an agent is a computer program that is capable of performing a flexible and independent action in typically dynamic and unpredictable domains (Luck, McBurney, Shehory, & Willmott, 2005). Agents are capable of performing actions and making decisions without the guidance of a human. Software agents emerged in the IT because of the ever-growing need for information processing, and the problems concerning dealing and working with large quantities of data.

Especially important is how agents act with other agents in the same environment, and the connections they form to find, refine and present the information in a best way. Agents certainly can do tasks better if they perform together, and that is why the multi-agent systems were developed.

The concept of an agent has become important in a diverse range of sub-disciplines of IT, including software engineering, networking, mobile systems, control systems, decision support, information recovery and management, e-commerce, and many others. Agents are now used in an increasingly wide number of applications — ranging from comparatively small systems such as web or e-mail filters to large, complex systems such as air-traffic control, that have a large dependency on fast and precise decision making.

Undoubtedly, the main contribution to the field of intelligent software agents came from the field of artificial intelligence (AI). The main focus of AI is to build intelligent entities and if these entities sense and act in some environment, then they can be considered agents (Russell & Norvig, 1995). Also, object-oriented programming (Booch, 2004), concurrent object-based systems (Agha, Wegner, and Yonezawa, 1993), and human-computer interaction (Maes, 1994) are fields that constantly drive forward the development of agents.

BACKGROUND

Although the term 'agent' is widely used, by many people working in closely related areas, it defies attempts to produce a single universally accepted definition. One of the most broadly used definitions states that *"an agent is an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objectives"* (Wooldridge and Jennings, 1995).

There are three main concepts in this definition: *situatedness, autonomy,* and *flexibility:*

- *Situatedness* means that an agent is situated in some environment and that it receives sensory input and performs actions which change that environment in some way.
- *Autonomy* is the ability of an agent to act without the direct intervention of humans. It has control over its own actions and over its internal state. Also, the autonomy implies the capability of learning from experience.

- *Flexibility* means that the agent is able to perceive its environment and respond to changes in a timely fashion; it should be able to exhibit opportunistic, goal-directed behaviour and take the initiative whenever appropriate. In addition, an agent should be able to interact with other agents and humans, thus to be 'social'.

For some researchers - particularly those interested in AI - the term `agent' has a stronger and more specific meaning than that sketched out above. These researchers generally mean an agent to be a computer system that, in addition to having the properties identified above, is either conceptualized or implemented using concepts that are more usually applied to humans. For example, it is quite common in AI to characterize an agent using mentalistic notions, such as knowledge, belief, intention, and obligation (Wooldridge & Jennings, 1995).

INTELLIGENT SOFTWARE AGENTS

Agents and Environments

An agent collects its percepts through its sensors, and acts upon the environment through its actuators. Thus, the agent is proactive. Its actions in any moment depend on the whole sequence of these inputs up to that moment. A decision tree for every possible percept

Figure 1. Agent and environment
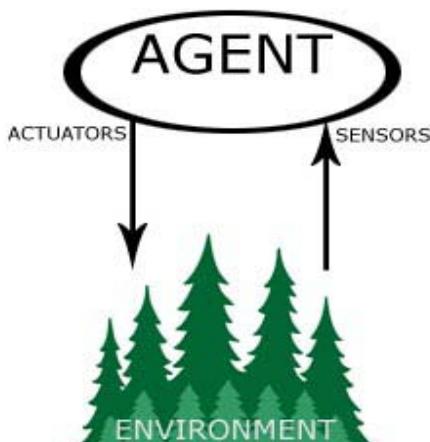
sequence of an agent would completely define the agent's behavior. This would define the function that maps any sequence of percepts to the concrete action – *the agent function*. The program that defines the agent function is called the *agent program*. So, the agent function is a formal description of the agent's behavior, and the agent program is a concrete implementation of that formalism. (Krcadinac, Stankovic, Kovanovic & Jovanovic, 2007)

To implement all this, we need to have a computing device with appropriate sensors and actuators on which the agent program will run. This is called *agent architecture*. So, an agent is essentially made of two components: the agent architecture and the agent program.

Also, as Russell and Norvig (1995) specify, one of the most sought after characteristics of an agent is its *rationality*. An agent is rational if it always does the action that will lead to the most successful outcome. The rationality of an agent depends on (a) the performance measure that defines what is a good action and what is a bad action, (b) the agent's knowledge about the environment, (c) the agent's available actions, and (d) the agent's percept history.

The Types of Agents

There are several basic types of agents with respect to their structure (Russell & Norvig, 1995):

1. The simplest kind of agents are the *simple reflex agents*. Such an agent only reacts to its current percept, completely ignoring its percept history. When a new percept is received, a rule that maps that percept to an action is activated. Such rules are known as *condition-action rules*.
2. *Model-based reflex agents* are more powerful agents, because they maintain some sort of internal state of the environment that depends on the percept history. For maintaining this sort of information, an agent must know how the environment evolves, and how its actions affect the environment.
3. *Goal-based agents* have some sort of goal information that describes desirable states of the world. Such an agent's decision making process is fundamentally different, because when a goal-based agent is considering performing an action it is asking itself "would this action make me

happy?" along with the standard "what this action will have as a result?".

4. *Utility-based agents* use a utility function that maps each state to a number that represents the degree of happiness. They are able to perform rationally even in the situations when there are conflicting goals, as well as when there are several goals that can be achieved, but none with certainty.

5. *Learning agents* do not have a priori knowledge of the environment, but *learn* about it. This is beneficial because these agents can operate in unknown environments and to a certain degree facilitates the job of developers because they do not need to specify their whole knowledge base.

Multi-Agent Systems

Multi-Agent Systems (MAS) are systems composed of multiple autonomous components (agents). They historically belong to *Distributed Artificial Intelligence* (Bond & Gasser, 1998). MAS can be defined as a loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of a single problem solver (Durfee and Lesser, 1989). In a MAS, each agent has incomplete information or capabilities for solving the problem and thus has a limited viewpoint. There is no global system control, the data is decentralized and the computation is asynchronous.

In addition to MAS, there is also the concept of a *multi-agent environment*, which can be seen as an environment that includes more than one agent. Thus, it can be cooperative, or competitive, or a combined one, and creates a setting where agents need to interact (socialize) between each other, either to achieve their individual objectives, or to manage the dependencies that follow from being situated in a common environment. These interactions range from simple semantic interoperation (exchanging comprehensible communications), client-server interactions (the ability to request that a particular action is performed), to rich social interactions (the ability to cooperate, coordinate, and negotiate about a course of action).

Because of the issues due to heterogeneous nature of agents involved in communication (e.g., finding one another), there is also a need for *middle-agents*, which cover cooperation among agents and connect

service providers with service requesters in the agent world. These agents are useful in various roles, such as *matchmakers* or *yellow page agents* that collect and process service offers ("advertisements"), *blackboard* agents that collect requests, and *brokers* that process both (Sycara, Decker, & Williamson, 1997). There are several alternatives to middle agents, such as Electronic Institutions – a framework for Agents' Negotiation which seeks to incorporate organizational concepts into multi-agent systems. (Rocha and Oliveira, 2001)

Communication among agents is achieved by exchanging messages represented by mutually understable language (syntax) and containing mutually understandable semantics. In order to find a common ground for communication, an *agent communication language* (*ACL*) should be used to provide mechanisms for agents to negotiate, query, and inform each other. The most important such languages today are *KQML* (Knowledge Query and Manipulation Language) (ARPA Knowledge Sharing Initiative, 1993) and *FIPA ACL* (FIPA, 1997).

AGENT APPLICABILITY

There are great possibilities for applying multi-agent systems to solving different kinds of practical problems.

- *Auction negotiation model*, as a form of communication, enables a group of agents to find good solutions by achieving agreement and making mutual compromises in case of conflicting goals. Such an approach is applicable to trading systems, where agents act on behalf of buyers and sellers. Financial markets, as well as scheduling, travel arrangement, and fault diagnosing also represent applicable fields for agents.

- Another very important field is *information gathering*, where agents are used to search through diverse and vastly different information sources (e.g., World Wide Web) and acquire relevant information for their users. One of the most common domains is Web browsing and search, where agents are used to adapt the content (e.g., search results) to the users' preferences and offer relevant help in browsing.

- *Process control software systems* require various kinds of automatic (autonomous) control and re-

action for its processes (e.g. production process). Reactive and responsive, agents perfectly fit the needs of such a task. Example domains in this field include: production process control, climate monitoring, spacecraft control, and monitoring nuclear power plants.

- *Artificial life* studies the evolution of agents, or populations of computer simulated life forms in artificial environments. The goal is to study phenomena found in real life evolution in a controlled manner, hopefully to eliminate some of the inherent limitations and cruelty of evolutionary studies using live animals.

- Finally, *intelligent tutoring systems* often include pedagogical agents, which represent software entities constructed to present the learning content in a user-friendly fashion and monitor the user's progress through the learning process. These agents are responsible for guiding the user and suggesting additional learning topics related to the user's needs (Devedzic, 2006).

Some of the more specific examples of intelligent agent applications include Talaria System, military training, and Mobility Agents. T a l a r i a System (The Autonomous Lookup And Report Internet Agent System) is a multi-agent system, developed for academic purposes at the University of Belgrade, Serbia. It was built as a solution to the common problem of gathering information from diverse Web sites that do not provide RSS feeds for news tracking. The system was implemented using the JADE modeling framework in Java. (Stankovic, Krcadinac, Kovanovic & Jovanovic, 2007) Talaria System is using the advantages of human-agent communication model to improve usability of web sites and to relieve users from annoying and repetitive work. The system provides each user with a personal agent, which periodically monitors the Web sites that the user expressed interest in. The agent informs its user about relevant changes, filtered by assumed user preferences and default relevance factors. Human-agent communication is implemented via email, so that a user can converse with her/his agent in natural language, whereas the agent heuristically interprets concrete instructions from the mail text (e.g., "monitor this site" or "kill yourself").

Simulation and modelling are extensively used in a wide range of military applications, from development, testing and acquisition of new systems and technologies, to operation, analysis and provision of training, and mission rehearsal for combat situations. The Human Variability in Computer Generated Forces (HV-CGF) project, undertaken on behalf of the UK's Ministry of Defence, developed a framework for simulating behavioral changes of individuals and groups of military personnel when subjected to moderating influences such as caffeine and fatigue. The project was built with the JACK Intelligent Agents toolkit, a commercial Java-based environment for developing and running multiagent applications. Each team member is a rational agent able to execute actions such as doctrinal and non-doctrinal behaviour tactics, which are encoded as JACK agent graphical plans. (Belecheanu et al., 2005)

Mobility Agents is an agent-based architecture that helps a person with cognitive disabilities to travel using public transportation. Agents are used to represent transportation participants (buses and travelers) and to enable notification of bus approaching and arrival. Information is passed to the traveler using a multimedia interface, via a handheld device. Customizable user profiles determine the most appropriate modality of interaction (voice, text, and pictures) based on the user's abilities (Repenning & Sullivan, 2003). This imposes a personal agent to take care that abstract goals, as "go home", are translated into concrete directions. To achieve this, an agent needs to collect information about user-specific locations and must be able to suggest the right bus for the particular user's current location and destination.

FUTURE TRENDS

Future looks bright for this technology as development is taking place within a context of broader visions and trends in IT. The whole growing field of IT is about to drive forward the R&D of intelligent agents. We especially emphasize *the Semantic Web, ambient intelligence, service oriented computing, Peer-to-peer computing* and *Grid Computing*.

The Semantic Web is the vision of the future Web based on the idea that the data on the Web can be defined and linked in such a way that it can be used by machines for the automatic processing and integration (Berners-Lee, Hendler, & Lassila, 2001). The key to achieving this is by augmenting Web pages with descriptions of their content in such a way that it is possible for

machines to reason automatically about that content. The common opinion is that the Semantic Web itself will be a form of intelligent infrastructure for agents, allowing them to "understand" the meaning of the data on the Web (Luck et al., 2005).

The concept of *ambient intelligence* describes a shift away from PCs to a variety of devices which are embedded in our environment and which are accessed via intelligent interfaces. It requires agent-like technologies in order to achieve autonomy, distribution, adaptation, and responsiveness.

Service oriented computing is where MAS could become very useful. In particular, this might involve web services, where the Quality Of Service demands are important. Each web service could be modeled as an agent, with dependencies, and then simulated for observed failure rates.

Peer-to-peer (P2P) computing, presenting networked applications in which every node is in some sense equivalent to all others, tends to become more complex in the future. Auction mechanism design, agent negotiation techniques, increasingly advanced approaches to trust and reputation, and the application of social norms, rules and structures - presents some of the agent technologies that are about to become relevant in the context of P2P computing.

Grid Computing is the high-performance agent-based computing infrastructure for supporting large-scale distributed scientific endeavour. The Grid provides a means of developing eScience applications, yet it also provides a computing infrastructure for supporting more general applications that involve large-scale information handling, knowledge management and service provision. The key benefit of Grid computing is flexibility – the distributed system and network can be reconfigured on demand in different ways as business needs change.

Some considerable challenges have still remained in the agent-based world, such as the lack of sophisticated software tools, techniques and methodologies that would support the specification, development, integration and management of agent systems.

CONCLUSION

Today, research and development in the field of intelligent agents is rapidly expanding. At its core is the concept of autonomous agents interacting with one another for their individual and/or collective benefit. A number of significant advances have been made over the past two decades in design and implementation of individual autonomous agents, and in the way in which they interact with one another. These concepts and technologies are now finding their way into commercial products and real-world software solutions. Future IT visions share the common need for agent technologies and prove that agent technologies will continue to be of vital importance. It is foreseeable that agents will become the integral part of informational technologies and artificial intelligence in the near future, and that is why they should be kept an eye on.

REFERENCES

Agha, G., Wegner, P., & Yonezawa, A. (Eds.). (1993). *Research directions in concurrent object-oriented programming*. Cambridge, MA: The MIT Press.

ARPA Knowledge Sharing Initiative. (1993). *Specification of the KQML agent-communication language – plus example agent policies and architectures*. Retrieved January 30, 2007, from http://www.csee.umbc.edu/kqml/papers/kqmlspec.pdf.

Barber, K. S., and Martin, C. E. (1999). *Agent Autonomy: Specification, Measurement, and Dynamic Adjustment*, Autonomy Control Software Workshop, Seattle, Washington.

Belecheanu, A. R., Luck, M., McBurney P., Miller T., Munroe, S., Payne T., & Pechoucek M. (2005). *Commercial Applications of Agents: Lessons, Experiences and Challenges*. (p. 2) Southampton, UK.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web, *Scientific American*, pp. 35-43.

Bond, A. H., & Gasser, L. (Eds.). (1998). *Readings in distributed artificial intelligence*. San Mateo, CA: Morgan Kaufmann Publishers.

Booch, G. (2004). *Object-oriented analysis and design (2nd ed.)*. MA: Addison-Wesley.

Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., & Orchard, D. (2004, February)., *Web services architecture. W3C working group note 11*. Retrieved January 30, 2007, from http://www.w3.org/TR/ws-arch/.

I

Devedzic, V. (2006). *Semantic web and education*. Berlin, Heidelberg, New York: Springer.

Durfee, E. H., & Lesser, V. (1989). Negotiating task decomposition and allocation using partial global planning. In L. Gasser, & M. Huhns (Eds.), *Distributed artificial intelligence: Volume II* (pp. 229–244) London: Pitman Publishing and San Mateo, CA: Morgan Kaufmann.

FIPA (1997). *Part 2 of the FIPA 97 specifications: Agent communication language*. Retrieved January 30, 2007, from http://www.fipa.org/specs/fipa00003/OC00003A.html.

Krcadinac, U., Stankovic, M., Kovanovic, V., & Jovanovic, J. (2007). Intelligent Multi-Agent Systems in: Carteli, A., & Palma, M. (Eds.). *Encyclopedia of Information Communication Technology*, Idea Group International Publishing, (forthcoming)

Luck, M., McBurney, P., Shehory, O., & Willmott, S. (2005). *Agent technology: Computing as interaction*. Retrieved January 30, 2007, from http://www.agentlink.org/roadmap/al3rm.pdf.

Maes, P.(1994) Agents that reduce work and information overload. *Communications of the ACM*, 37(7), 31–40.

Repenning, A., & Sullivan, J. (2003). The Pragmatic Web: Agent-Based Multimodal Web Interaction with no Browser in Sight, In G.W.M. Rauterberg, M. Menozzi, & J. Wesson, (Eds.), *Proceedings of the Ninth International Conference on Human-Computer Interaction* (pp. 212-219). Amsterdam, The Netherlands: IOS Press.

Rocha, A. P. & Oliveira, E. (2001) *Electronic Institutions as a framework for Agents' Negotiation and mutual Commitment*. In P. Brazdil, A. Jorge (Eds.), Progress in Artificial Intelligence (Proceedings of 10th EPIA), LNAI 2258, pp. 232-245, Springer.

Russell, S. J., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. New Jersey: Prentice-Hall.

Stankovic, M., Krcadinac, U., Kovanovic, V., & Jovanovic, J. (2007). An Overview of Intelligent Software

Agents in: Khosrow-Pour, M. (Ed.). *Encyclopedia of Information Science and Technology, 2nd Edition*, Idea Group International Publishing, (forthcoming)

Sycara, K., Decker, K., & Williamson, M. (1997). Middle-Agents for the Internet, In M. E. Pollack, (Ed.), *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (pp. 578-584). Morgan Kaufmann Publishers.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), pp. 115–152.

KEY TERMS

Actuators: Software component and part of the agent used as a mean of performing actions in the agent environment.

Agent Autonomy: Agent's active use of its capabilities to pursue some goal, without intervention by any other agent in the decision-making process used to determine how that goal should be pursued (Barber & Martin, 1999).

Agent Percepts: Every information that an agent receives trough it's sensors, about the state of the environment or any part of the environment.

Intelligent Software Agent: An encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives (Wooldridge & Jennings, 1995).

Middle-Agents: Agents that facilitate cooperation among other agents and typically connect service providers with service requesters.

Multi-Agent System (MAS): A software system composed of several agents that interact in order to find solutions of complex problems.

Sensors: Software component and part of the agent used as a mean of acquiring information about current state of the agent environment (i.e., agent percepts).

Intelligent Traffic Sign Classifiers

Raúl Vicen Bueno
University of Alcalá, Spain

Elena Torijano Gordo
University of Alcalá, Spain

Antonio García González
University of Alcalá, Spain

Manuel Rosa Zurera
University of Alcalá, Spain

Roberto Gil Pita
University of Alcalá, Spain

INTRODUCTION

The **Artificial Neural Network**s (**ANN**s) are based on the behavior of the brain. So, they can be considered as intelligent systems. In this way, the **ANN**s are constructed according to a brain, including its main part: the neurons. Moreover, they are connected in order to interact each other to acquire the followed intelligence. And finally, as any brain, it needs having memory, which is achieved in this model with their weights.

So, starting from this point of view of the **ANN**s, we can affirm that these systems are able to learn difficult tasks. In this article, the task to learn is to distinguish between different kinds of **traffic sign**s. Moreover, this **ANN** learning must be done for **traffic sign**s that are not in perfect conditions. So, the learning must be robust against several problems like rotation, translation or even vandalism. In order to achieve this objective, an intelligent extraction of information from the images is done. This stage is very important because it improves the performance of the **ANN** in this task.

BACKGROUND

The **Traffic Sign Classification** (**TSC**) problem has been studied many times in the literature. This problem is solved in (Perez, 2002, Escalera, 2004) using the correlation between the **traffic sign** and each element of a database, which involves large computational cost. In (Hsu, 2001), Matching Pursuit (MP) is applied in two

stages: training and testing. The training stage finds a set of the best MP filters for each **traffic sign**, while the testing one projects the unknown traffic sign to different MP filters to find the best match. This method also implies large computational cost, especially when the number of elements grows up. In recent works (Escalera, 2003, Vicen, 2005a, Vicen, 2005b), the use of **ANN**s is studied. The first one studies the combination of the Adaptive Resonance Theory with **ANN**s. It is applied to the whole image, where many **traffic sign**s can exist, which involves that the **ANN** complexity must be very high to recognize all the possible signs. In the last works, the **TSC** is constructed using a **preprocessing** stage before the **ANN**, which involves a computational cost reduction in the classifier.

TSCs are usually composed by two specific stages: the *detection* of **traffic sign**s in a video sequence or image and their *classification*. In this work we pay special attention to the classification stage. The performance of these stages highly depends on lighting conditions of the scene and the state of the **traffic sign** due to deterioration, vandalism, rotation, translation or inclination. Moreover, its perfect position is perpendicular to the trajectory of the vehicle, however many times it is not like that. Problems related to the **traffic sign** size are of special interest too. Although the size is normalized, we can find signs of different ones, because the distance between the camera and the sign is variable. So, the classification of a **traffic sign** in this environment is not easy.

The objective of this work is the study of different classification techniques combined with different **pre-processing**s to implement an intelligent **TSC** system. The **preprocessing**s considered are shown below and are used to reduce the classifier complexity and to improve its performance. The studied classifiers are the k-Nearest Neighbor (k-NN) and an **ANN** based method using **Multilayer Perceptron**s (**MLP**s). So, this work tries to find which are the best **preprocessing**s, the best classifiers and which combination of them minimizes the error rate.

INTELLIGENT TRAFFIC SIGN CLASSIFICATION

An intelligent **traffic sign** classification can be achieved taking into account two important aspects. The first one focus on the extraction of the relevant information of the input **traffic sign**s, which can be done adaptively or fixed. The second one is related with the classification core. From the point of view of this part, **ANN**s can play a great role, because they are able to learn from different environments. So, an intelligent combination of both aspects can lead us to the success in the classification of **traffic sign**s.

Traffic Sign Classification System Overview

The **TSC** system and the blocks that compose it are shown in figure 1. Once the *Video Camera* block takes a

video sequence, the *Image Extraction* block makes the video sequence easy to read and it is the responsible to obtain images. The *Sign Detection and Extraction Stage* extracts all the **traffic sign**s contained in each image and generates the small images called blobs, one per possible sign. Figure 1 also shows an example of the way this block works. The *Color Recognition Stage* is the responsible to discern among the different predominant color of the **traffic sign**: blue, red or others. Once the blob is classified according to its predominant color, the *TSC Stage* has the responsibility to recognize the exact type of signal, which is the aim of this work. This stage is divided in two parts: the **traffic sign preprocessing** stage and the **TSC** core.

Database Description

The database of blobs used to obtain the results presented in this work is composed of blobs with only noise and nine different types of blue **traffic sign**s, which belong to the international traffic code. Figure 2.a (Normal Traffic Signs) shows the different classes of **traffic sign**s considered in this work, which have been collected by the **TSC** system presented above. So, they present distortions due to the problems described in previous sections, which are shown in figure 2.b (Traffic Signs with problems). The problems caused by vandalism are shown in the example of class S_8. The problems related to the blob extraction in the *Sign Detection and Extraction Stage* (not a correct fit in the square image) are shown in the examples of classes S_2, S_4 and S_9. Examples of signs with problems of rotation, translation or inclination are those of classes S_4, S_6 and

Figure 1. Traffic sign classification system
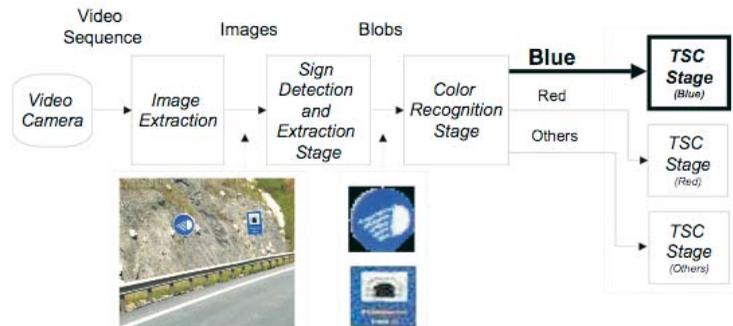
Figure 2. Noise and nine classes of international traffic signs: (a) Normal traffic signs and (b) Traffic signs with problems

<div align="center">(a)　　　　　　　　　　　　　　(b)</div>
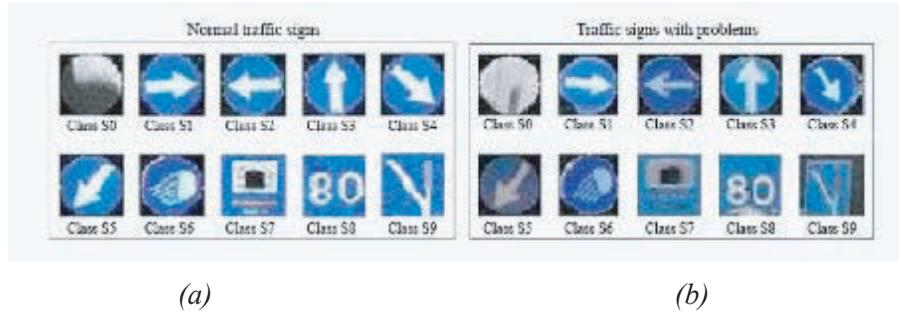
S_9. Finally, the difference of brightness is observed in both parts of figure 2. For example, when the lighting of the blob is high, the vertical row of the example of class S_3 is greater than horizontal row of the example of class S_2.

Traffic Sign Preprocessing Stage

Each blob presented at the input of the **TSC** stage contains information of the three-color components: red, green and blue. Each blob is composed of 31x31 pixels. So, the memory required for each blow is 2883 bytes. Due to the high quantity of data, the purpose of this stage is to reduce it and to limit the redundancy of information, in order to improve the **TSC** performance and to reduce the **TSC** core computational cost.

The first **preprocessing** made in this stage is the transformation of the color blob (3x31x31) to a gray scale blob (31x31) (Paulus, 2003). Consider for the next explanation that **M** is a general bidimensional matrix that contains either the gray scale blob or the output of one of the next **preprocessing**s:

- *Median filter* (MF) (Abdel, 2004). It is applied to each pixel of **M**. A block of *nxn* elements that surrounds a pixel of **M** is taken, which is sorted in a linear vector. The median value of this vector is selected as the value of the processed pixel. This

preprocessing is usually used to reduce the noise in an image.

- *Histogram equalization* (HE). It tries to enhance the contrast of **M**. The pixels are transformed according to a specified image **histogram** (Paulus, 2003). This equalization is usually used to improve the dynamic range of **M**.

- *Vertical* (VH) and *horizontal* (HH) *histograms* (Vicen, 2005a, Vicen, 2005b). They are computed with

$$vh_i = \frac{1}{31} \sum_{j=1}^{31} \left(m_{i,j} > T \right) \ , \quad i = 1,2,\ldots,31 \tag{1}$$

$$hh_j = \frac{1}{31} \sum_{i=1}^{31} \left(m_{i,j} > T \right) \ , \quad j = 1,2,\ldots,31 \tag{2}$$

respectively, where $m_{i,j}$ is the element of the *i-th* row and *j-th* column of the matrix **M** and **T** is the fixed or adaptive threshold of this **preprocessing**. If **T** is fixed, it is established at the beginning of the **preprocessing**, but if **T** is adaptive, it can be calculated with the Otsu method (Ng, 2004) or with the mean value of the blob, so both methods are **M**-dependent. vh_i corresponds to the ratio of values of column *j-th* that are greater than **T** and hh_j corresponds to the ratio of values of row *i-th* that are greater than **T**.

Traffic Sign Classification Core

TSC can be formulated as a multiple hypothesis test. Consider that $P(D_i|S_j)$ is the probability of deciding in favor of S_i (decision D_i) when the true hypothesis is S_j, $C_{i,j}$ is the cost associated with this decision and $P(S_j)$ is the prior probability of hypothesis S_j. Then the objective is to minimize a risk function that is given as the average cost $\overline{C}$, which is defined in (3) for L hypothesis.

$$\overline{C} = \sum_{i=1}^{L}\sum_{j=1}^{L} C_{i,j} P(D_i / S_j) P(S_j)$$

(3)

The classifier performance can be given as the *total error rate* (P_e) and the *total correct rate* ($P_c = 1 - P_e$) for all the hypothesis (classes).

Traffic Sign Classification Core Based on Statistical Methods: The *k*-NN

The k-NN approach is a widely-used statistical method (Kisienski, 1975) applied in classification tasks. It assumes that the training set contains M_i points of class S_i and M points in total, so

$$\sum_i M_i = M.$$

Then a hypersphere around the observation point $\mathbf{x}$ is taken, which encompasses k points irrespective of their class label. Suppose this sphere, of volume V, contains k_i points of class S_i, then

$$p(x / S_i) \approx \frac{k_i}{M_i V}$$

(4)

provides an approximation to this class-conditional density. The unconditional density can be estimated using

$$p(x) \approx \frac{k}{MV}$$

(5)

while the priors can be estimated using

$$p(S_i) \approx \frac{M_i}{M}.$$

(6)

Then applying Bayes' theorem (Bishop, 1995), we obtain:

$$p(S_i / x) = \frac{p(x / S_i)p(S_i)}{p(x)} \approx \frac{k_i}{k}.$$

(7)

Thus, to minimize the probability of misclassifying $\mathbf{x}$, it should be assigned to the class S_i for which the ratio k_i/k is highest. The way to apply this method consists in comparing each $\mathbf{x}$ of the test set with all the training set patterns and deciding which class S_i is the most appropriate one. k denotes the number of patterns that take part in the final decision of classifying $\mathbf{x}$ in class S_i. When a draw exists in the majority voting, the decision is taking using the class of the nearest pattern. So, the results for $k=1$ and $k=2$ are the same.

Traffic Sign Classification Core Based on Neural Networks: The MLP

The Perceptron was developed by F. Rosenblatt (Rosenblatt, 1962) in the 1960s for optical character recognition. The Perceptron has multiple inputs fully connected to an output layer with multiple outputs. Each output y_i is the result of applying a linear combination of the inputs to a non-linear function called activation function. **MLP**s (Haykin, 1999) extend the Perceptron by cascading one or more extra layers of processing elements. These layers are called hidden layers, since their elements are not connected directly to the external world. The expression $I/H_1/.../H_h/O$ denotes an **MLP** with I inputs (size of the observation vector $\mathbf{x}$), h hidden layers with H_h neurons in each one and O outputs (size of the classification vector $\mathbf{y}$).

Cybenko's theorem (Cybenko, 1989) states that any continuous function $f \bullet \mathfrak{R}^n \to \mathfrak{R}$ can be approximated with any degree of precision by log-sigmoidal functions. Therefore, MLPs using the log-sigmoidal activation function for each neuron are selected.

Gradient descent with momentum and adaptive learning rate backpropagation algorithm is used to train the **MLP**s, where the *Mean Square Error* (MSE)

criterion is minimized. Moreover, cross-validation is used in order to reduce generalization problems.

RESULTS

The database considered for the experiments is composed of 235 blobs of ten different classes: noise (S_0) and nine classes of **traffic sign**s (S_1-S_9). The database has been divided in three sets: train, validation and test, which are composed of 93, 52 and 78 blobs, respectively, being preprocessed before they are presented to the **TSC** core. The first one is used as the training set for the k-NN and the **MLP**s. The second one is used to stop the **MLP** training algorithm (Bishop, 1995, Haykin, 1999) according to the cross-validation applied during the training. And the last one is used to evaluate the performance of the k-NN and the **MLP**s. Experimental environments characterized by a large dimensional space and a small data set pose generalization problems. For this reason, the **MLP**s training is repeated 10 times with different weights initialization each time and the best **MLP** in terms of P_e estimated with the validation set is selected.

Once the color blobs are transformed to gray scale, three different *combinations of preprocessing* (CPPs) are applied, so each CPP output is 62 elements:

- The first combination (CPP1) applies the VH and HH with an adaptive threshold T calculated with the mean of the blob.
- The second combination (CPP2) applies, in this order, the HE and the VH and HH with an adaptive threshold T calculated with the Otsu method.
- The third combination (CPP3) applies, in this order, the MF, the HE and the VH and HH with a fixed threshold (T=185).

For the **TSC** core based on the k-NN, a study of the k parameter is made for the different CPPs considered in the experiments (table 1). The lowest error rate is achieved with CPP3 and k=1, which performance is P_e=6,4% (P_c=93,6%).

For the **TSC** core based on **MLP**s, a study of the number of hidden layers (h) and the number of neurons in each one (H_h) is done.

For the case of one hidden layer (h=1), table 2 shows the results for the different CPPs. In this case, the best

performance is obtained with the CPP3 and an **MLP** of 62/62/10, where its error rate is P_e=2,6% (P_c=97,4%). The CPP2 achieves good performances but they are always lower than in the case of using the CPP3. The use of CPP1 with MLPs achieves the worst results of the three cases under study.

The study of the **TSC** core based on an MLP with two hidden layers (h=2) (table 3) shows that the best combination of the CPPs and [H_1,H_2] for the **MLP** is CPP3 and [H_1=70,H_2=20], respectively. In this case, the best performance achieved is P_e=1,3% (P_c=98,7%). As occurs for **MLP**s with one hidden layer, the best CPP is the third one and the worst one is the first one.

FUTURE TRENDS

New innovations can be achieved in this research area. The new trends try to improve the **preprocessing** techniques. In this case, advance signal processing can be applied to **TSC**. On the other hand, other **TSC** cores can be used. For instance, classifiers based on Radial Basis Function or Support Vector Machines (Maldonado, 2007) can be applied. Finally, optimization techniques, like **Genetic Algorithm**s, have an important role in this research area to find which is the best selection of **preprocessing**s of a bank of them.

CONCLUSION

The performances of all the **TSC** designs are quite good, even though when the problems of deterioration, vandalism, rotation, translation, inclination, not a correct fit in the 31x31 blob and variation in size exist in the blobs.

Several combinations of **preprocessing**s are used. The best one applies, in this order, the *median filter*, the *histogram equalization* and the *vertical* and *horizontal histograms* with a fixed threshold (*T*=185).

Concerning the type of classifier, the best **TSC**s are always achieved with **MLP**s. Moreover, the best results are achieved by **MLP**s of two hidden layers. The P_e reduction of the **TSC** core based on a 62/70/20/10 **MLP** (P_e=1,3%) is of 1,3% with respect to the best one achieved with only one hidden layer **MLP** (62/62/10) and 5,1% with respect to the best k-NN (k=1) achieved.

I

Table 1.P_e(%) versus k parameter for each TSC based on different CPPs and k-NN

k	1	3	4	5	6	7	8	9	10	11	12
CPP1	29,5	30,8	29,5	29,5	32,0	30,8	30,8	28,2	**25,6**	25,6	25,6
CPP2	19,2	17,9	**14,1**	16,7	14,1	15,4	19,2	16,7	17,9	19,2	19,2
CPP3	**6,4**	9,0	9,0	11,5	12,8	12,8	12,8	12,8	12,8	12,8	10,3

Table 2. P_e(%) versus H_1 parameter for each TSC based on different CPPs and MLPs of sizes (62/H_1/10)

H_1	6	14	22	30	38	46	54	62	70	78	86
CPP1	24,4	17,9	17,9	15,4	18,9	16,7	**14,1**	17,9	19,2	17,9	15,4
CPP2	21,8	14,1	14,1	14,1	12,8	10,3	11,5	10,3	12,8	**9,0**	11,5
CPP3	12,8	3,8	5,1	3,8	3,8	5,1	3,8	**2,6**	5,1	5,1	3,8

Table 3. P_e(%) versus [H_1, H_2] parameters for each TSC based on different CPPs and MLPs of sizes (62/H_1/H_2/10)

H_1	10	10	15	15	25	25	40	40	60	60	70	70
H_2	6	8	5	7	8	10	15	20	18	25	20	30
CPP1	28,2	24,4	23,1	25,6	19,2	19,2	19,2	19,2	17,9	17,9	**15,4**	15,4
CPP2	25,6	25,6	26,9	23,1	17,9	20,5	16,7	11,5	12,8	11,5	12,8	**9,0**
CPP3	15,4	10,3	15,4	12,8	7,7	5,1	6,4	5,1	5,1	5,1	**1,3**	5,1

REFERENCES

Abdel-Dayem, A.R., Hamou, A.K., & El-Sakka, M.R. (2004). Novel Adaptive Filtering for Salt-and-Pepper Noise Removal from Binary Document Images. *Lecture Notes in Computer Science*. (3212), 191-199.

Bishop, C.M. (1995). *Neural networks for pattern recognition.* Oxford University Press Inc.

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems*. (2), 303-314.

Escalera, A. de la, et. al. (2004). Visual Sign Information Extraction and Identification by Deformable Models for Intelligent Vehicles. *IEEE Trans. on Intelligent Transportation Systems*. (5) 2, 57-68.

Escalera, A. de la, et. al. (2003). Traffic Sign Recognition and Analysis For Intelligent Vehicles. *Image and Vision Computing*. (21), 247-258.

Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation (Second Edition)*. Prentice-Hall.

Hsu, S.H., & Huang, C.L. (2001). Road Sign Detection and Recognition Using Matching Pursuit Method. *Image and Vision Computing*. (19), 119-129.

Kisienski, A. A., et al. (1975). Low-frequency Approach to Target Identification. *Proc. IEEE*. (63), 1651-1659.

Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H., & Lopez-Ferreras, F. (2007). Road-Sign Detection and Recognition Based on Support Vector Machines. *IEEE Trans. on Intelligent Transportation Systems*. (8) 2, 264-278.

Ng, H.F. (2004). Automatic Thresholding for Defect Detection. *IEEE Proc. Third Int. Conf. on Image and Graphics*. 532-535.

Paulus, D.W.R., & Hornegger, J. (2003). *Applied Pattern Recognition (4th Ed.): Algorithms and Implementation in C++*. Vieweg.

Pérez, E., & Javidi, B. (2002). Nonlinear Distortion-Tolerant Filters for Detection of Road Signs in Background Noise. *IEEE Trans. on Vehicular Technology*. (51) 3, 567-576.

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan books.

Vicen-Bueno, R., Gil-Pita, R., Rosa-Zurera, M., Utrilla-Manso, M., & López-Ferreras, F. (2005a). Multilayer Perceptrons Applied to Traffic Sign Recognition Tasks. *Lecture Notes in Computer Science*. (3512), 865-872.

Vicen-Bueno, R., Gil-Pita, R., Jarabo-Amores, M. P., & López-Ferreras, F. (2005b). Complexity reduction in Neural Networks applied to Traffic Sign Recognition tasks. *13th European Signal Processing Conference. EUSIPCO 2005*.

KEY TERMS

Artificial Neural Networks (ANNs): A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Backpropagation Algorithm: Learning algorithm of ANNs, based on minimizing the error obtained from the comparison between the ANN outputs after the application of a set of network inputs and the desired outputs. The update of the weights is done according to the gradient of the error function evaluated in the point of the input space that indicates the input to the ANN.

Classification: The act of distributing things into classes or categories of the same type.

Detection: The perception that something has occurred or some state exists.

Information Extraction: Obtention of the relevant aspects contained in data. It is commonly used to reduce the input space of a classifier.

Pattern: Observation vector that for its relevance is considered as an important example of the input space.

Preprocessing: Operation or set of operations applied to a signal in order to improve some aspects of it.

I

Interactive Systems and Sources of Uncertainties

Qiyang Chen
Montclair State University, USA

John Wang
Montclair State University, USA

INTRODUCTION

Today's e-commerce environment requires that interactive systems exhibit abilities such as autonomy, adaptive and collaborative behavior, and inferential capability. Such abilities are based on the knowledge about users and their tasks to be performed (Raisinghani, Klassen and Schkade, 2001). To adapt users' input and tasks an interactive system must be able to establish a set of assumptions about users' profiles and task characteristics, which is often referred as user models. However, to develop a user model an interactive system needs to analyze users' input and recognize the tasks and the ultimate goals users trying to achieve, which may involve a great deal of uncertainties.

Uncertainty refers to a set of values about a piece of assumption that cannot be determined during a dialog session. In fact, the problem of uncertainty in reasoning processes is a complex and difficult one. Information available for user model construction and reasoning is often uncertain, incomplete, and even vague. The propagation of such data through an inference model is also difficult to predict and control. Therefore, the capacity of dealing with uncertainty is crucial to the success of any knowledge management system.

Currently, a vigorous debate is in progress concerning how best to represent and process uncertainties in knowledge based systems. This debate carries great importance because it is not only related to the construction of knowledge based system but also focuses on human thinking in which most decisions are made under conditions of uncertainty. This chapter presents and discusses uncertainties in the context of user modeling in interactive systems. Some elementary distinctions between different kinds of uncertainties are introduced. The purpose is to provide an analytical overview and perspective concerning how and where uncertainties arise and the major methods that have been proposed to cope with them.

Sources of Uncertainties

The user model based interactive systems face the problems of uncertainty in the reference rule, the facts, and representation languages. There is no widely accepted definition about the presence of uncertainty in user modeling. However, the nature of uncertainty in a user model can be investigated through its origin. Uncertainty can arise from a variety of sources. Several authors have emphasized the need for differentiating among the types and sources of uncertainty. Some of the major sources are as follows:

(1) The imprecise and incomplete information obtained from the user's input. This type of source is related to the reliability of information, which involves the following aspects:

- Uncertain or imprecise information exists in the factual knowledge (Dutta, 2005). The contents of a user model involve uncertain factors. For instance, the system might want to assert "It is not likely that this user is a novice programmer." This kind of assertion might be treated as a piece of knowledge. But it is uncertain and seems difficult to find a numerical description for the uncertainty in this statement (*i.e.*, no appropriate sample space in which to give this statement statistical meaning, if a statistical method is considered for capturing the uncertainty).
- The default information often brings uncertain factors to inference processes (Reiter, 1980). For example, the stereotype system carries extensive default assumptions about a user. Some assump-

tions may be subject to change as interaction progresses.

- Uncertainty occurs as a result of ill-defined concepts in the observations or due to inaccuracy and poor reliability of the measurement (Kahneman and Tversky, 1982). For example, a user's typing speed could be considered as a measurement for a user's file editing skill. But for some applications it may be questionable.
- The uncertain exception to general assumptions for performing some actions under some circumstances can cause conflicts in reasoning processes.

(2) Inexact language by which the information is conveyed. The second source of uncertainty is caused by the inherent imprecision or inexactness of the representation languages. The imprecision appears in both natural languages and knowledge representation language. It has been proposed to classify three kinds of inexactness in natural language (Zwick, 1999). The first is generality, in which a word applies to a multiplicity of objects in the field or reference. For example, the word "table" can apply to objects differing in size, shape, materials, and functions. The second kind of linguistic exactness is ambiguity, which appears when a limited number of alternative meanings have the same phonetic form (*e.g.*, bank). The third is vagueness, in which there are no precise boundaries to the meaning of the word (*e.g.*, old, rich).

In knowledge representation languages employed in user modeling systems, if rules are not expressed in a formal language, their meaning usually cannot be interpreted exactly. This problem has been partially addressed by the theory of approximate reasoning. Generally, a proposition (*e.g.,* fact, event) is uncertain if it involves a continuous variable. Note that an exact assumption may be uncertain (*e.g.*, the user is able to learn this concept), and an assumption that is absolutely certain may be linguistically inexact (e.g. the user is familiar with this concept).

(3) Aggregation or summarization of information. The third type of uncertainty source arises from aggregation of information from different knowledge sources or expertise (Bonissone and Tong, 2005). Aggregating information brings several potential problems that are discussed in (Chen and Nocio 1997).

(4) Deformation while transferring knowledge. There might be no semantic correspondence between one representation language to another. It is possible that there is even no appropriate representation for certain expertise, for example, the measurement of user's mental workload. This makes the deformation of transformation inevitable. In addition, human factors greatly affect the procedure of information translation. Several tools that use cognitive models for knowledge acquisition have been presented (Jacobson and Freiling, 1988).

CONCLUSION

Generally, uncertainty affects the performance of an adaptive interface in the following aspects and obviously, the management of uncertainty must address all of the following aspects (Chen and Norcio, 2001).

- How to determine the degree to which the premise of a given rule has been satisfied.
- How to verify the extent to which external constraints have been met.
- How to propagate the amount of uncertain information through triggering of a given rule.
- How to summarize and evaluate the findings provided by various rules or domain expertise.
- How to detect possible inconsistencies among the various sources and,
- How to rank different alternatives or different goals.

REFERENCES

Barr, A. and Feigenbaum, E. A., *The Handbook of Artificial Intelligence 2*. Los Altos, Kaufmann , 1982.

Bhatnager, R. K. and Kanal, L. N., "Handling Uncertainty Information: A Review of Numeric and Nonnumeric Methods," *Uncertainty in Artificial Intelligence*, Kanal, L. N. and Lemmer, J. F. (ed.), pp2-26, 1986.

Bonissone, P. and Tong, R. M., "Editorial: Reasoning with Uncertainty in Expert Systems," *International Journal of Man-Machine Studies*, Vol. 30, 69-111 (2005)

I

Buchanan, B. and Smith, R. G. Fundamentals of Expert Systems, *Ann. Rev., Computer Science*, Vol. 3, pp. 23-58, 1988.

Chen, Q. and Norcio, A.F. "Modeling a User's Domain Knowledge with Neural Networks," *International Journal of Human-Computer Interaction*, Vol. 9, No. 1, pp. 25-40, 1997.

Chen, Q. and Norcio, A.F. "Knowledge Engineering in Adaptive Interface and User Modeling," *Human-Computer Interaction: Issues and Challenges,* (ed.) Chen, Q. Idea Group Pub. 2001.

Cohen, P. R. and Grinberg, M. R., " A Theory of Heuristic Reasoning about Uncertainty, AI Magazine, Vol. 4(2), pp. 17-23, 1983.

Dempster, A. P., "Upper and Lower Probabilities Induced by a Multivalued mapping," *The Annuals of Mathematical Statistics* , Vol. 38(2), pp. 325-339, 1967.

Dubois, D. and Prade, H., "Combination and Propagation of Uncertainty with Belief Functions -- A Reexamination," *Proc. of 9th International Joint Conference on Artificial Intelligence,* pp. 111-113, 1985.

Dutta, A., "Reasoning with Imprecise Knowledge in Expert Systems," Information Sciences, Vol. 37, pp. 2-24, 2005.

Doyle, J., "A Truth Maintenance System," *AI,* Vol. 12, 1979, pp. 231-272.

Garvey, T. D., Lowrance, J. D. and Fischer, M. A. "An Inference Technique for Integrating Knowledge from Disparate Source," *Proc. of the 7th International Joint Conference on AI,* Vancouver, B. C. pp. 319-325, 1982

Heckerman, D., "Probabilistic Interpretations for MYCIN's Certainty actors," *Uncertainty in Artificial Intelligence,*(ed.). Kanal, L. N. and Lemmer, J. F. , 1986

Jacobson, C. and Freiling, M. J. "ASTEK: A Mulitparadigm Knowledge Acquisition tool for complex structured Knowledge." *International. Journal of Man-Machine Studies,* Vol. 29, 311-327. 1988.

Kahneman, D. and Tversky, A (1982). Variants of Uncertainty, Cognition, 11, 143-157.

McDermott, D. and Doyle, J., "Non-monotonic Logic," *AI* Vol. 13, pp. 41-72. (1980).

Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publisher, San Mateo, CA ,1988.

Pearl, J., "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach," *Proc. of the 2nd National Conference on Artificial Intelligence*, IEEE Computer Society, pp. 1-12, 1985.

Pednault, E. P. D., Zucker, S. W. and Muresan, L.V., "On the Independence Assumption Underlying Subjective Bayesian Updating, " *Artificial Intelligence*, 16, pp. 213-222. 1981

Raisinghani, M., Klassen, C. and Schkade, L. "Intelligent Software agents in Electonic Commerce: A Socio-Technical Perspective," *Human-Computer Interaction: Issues and Challenges,* (ed.) Chen, Q. Idea Group Pub. 2001.

Reiter, R., "A Logic for Default Reasoning," *Artificial Intelligence*, Vol. 13, 1980 pp. 81-132.

Rich, E., "User Modeling via Stereotypes," *Cognitive Sciences,* Vol. 3 1979, pp. 329-354.

Shafer, G., *A Mathematical Theory of Evidence,* Priceton University Press, 1976.

Zadeh, L. A., " Review of Books : A Mathematical Theory of Evidence," *AI Magazine.*, 5(3), 81-83. 1984

Zadeh, L. A. "Knowledge Representation in Fuzzy Logic," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 1, No. 1, pp. 89-100, 1989.

Zwick, R., "Combining Stochastic Uncertainty and Linguistic Inexactness: Theory and Experimental Evaluation of Four Fuzzy Probability Models," *Int. J. Man-Machine Studies,* Vol. 30, pp. 69-111, 1999.

KEY TERMS

Interactive System: A system that allows dialogs between the computer and the user.

Knowledge Based Systems: A computer system that programmed to imitate human problem-solving

by means of artificial intelligence and reference to a database of knowledge on a particular subject.

Knowledge Representation: The notation or formalism used for coding the knowledge to be stored in a knowledge-based system.

Stereotype: A set of assumptions based on conventional, formulaic, and simplified conceptions, opinions about a user, which is created by an interactive system.

Uncertainties: A potential deficiency in any phase or activity of the modeling process that is due to the lack of knowledge

User Model: A set of information an interactive system infers or collects, which is used to characterize a user's tasks, goals, domain knowledge and preferences, etc. to facilitate human computer interaction.

Intuitionistic Fuzzy Image Processing

Ioannis K. Vlachos
Aristotle University of Thessaloniki, Greece

George D. Sergiadis
Aristotle University of Thessaloniki, Greece

INTRODUCTION

Since its genesis, fuzzy sets (FSs) theory (Zadeh, 1965) provided a flexible framework for handling the indeterminacy characterizing real-world systems, arising mainly from the imprecise and/or imperfect nature of information. Moreover, fuzzy logic set the foundations for dealing with reasoning under imprecision and offered the means for developing a context that reflects aspects of human decision-making. Images, on the other hand, are susceptible of bearing ambiguities, mostly associated with pixel values. This observation was early identified by Prewitt (1970), who stated that "a pictorial object is a fuzzy set which is specified by some membership function defined on all picture points", thus acknowledging the fact that "some of its uncertainty is due to degradation, but some of it is inherent". A decade later, Pal & King (1980) (1981) (1982) introduced a systematic approach to fuzzy image processing, by modelling image pixels using FSs expressing their corresponding degrees of brightness. A detailed study of fuzzy techniques for image processing and pattern recognition can be found in Bezdek et al and Chi et al (Bezdek, Keller, Krisnapuram, & Pal, 1999) (Chi, Yan, & Pham, 1996).

However, FSs themselves suffer from the requirement of *precisely* assigning degrees of membership to the elements of a set. This constraint raises some of the flexibility of FSs theory to cope with data characterized by uncertainty. This observation led researchers to seek more efficient ways to express and model imprecision, thus giving birth to higher-order extensions of FSs theory.

This article aims at outlining an alternative approach to digital image processing using the apparatus of Atanassov's intuitionistic fuzzy sets (A-IFSs), a simple, yet efficient, generalization of FSs. We describe heuristic and analytic methods for analyzing/synthesizing images to/from their intuitionistic fuzzy components and discuss the particular properties of each stage of the process. Finally, we describe various applications of the intuitionistic fuzzy image processing (IFIP) framework from diverse imaging domains and provide the reader with open issues to be resolved and future lines of research to be followed.

BACKGROUND

From the very beginning of their development, FSs intrigued researchers to apply the flexible fuzzy framework in different domains. In contrast with ordinary (crisp) sets, FSs are defined using a characteristic function, namely the *membership function*, which maps elements of a universe to the unit interval, thereby attributing values expressing the *degree of belongingness* with respect to the set under consideration. This particular property of FSs theory was exploited in the context of digital image processing and soon turned out to be a powerful tool for handling the inherent uncertainty carried by image pixels. The importance of fuzzy image processing was rapidly acknowledged by both theoreticians and practitioners, who exploited its potential to perform various image-related tasks, such as contrast enhancement, thresholding and segmentation, de-noising, edge-detection, and image compression.

However, and despite their vast impact to the design of algorithms and systems for real-world applications, FSs are not always able to directly model uncertainties associated with imprecise and/or imperfect information. This is due to the fact that their membership functions are themselves crisp. These limitations and drawbacks characterizing most ordinary fuzzy logic systems (FLSs) were identified and described by Mendel & Bob John (2002), who traced their sources back to the uncertainties that are present in FLSs and arise from various factors. The very meaning of words that are used in the antecedents and consequents of FLSs can

be uncertain, since some words may often mean different things to different people. Moreover, extracting the knowledge from a group of experts who do not all agree, leads in consequents having a histogram of values associated with them. Additionally, data presented as inputs to an FLS, as well as data used for its tuning, are often noisy, thus bearing an amount of uncertainty. As a result, these uncertainties translate into additional uncertainties about FS membership functions. Finally, Atanassov et al. (Atanassov, Koshelev, Kreinovich, Rachamreddy & Yasemis, 1998) proved that there exists a fundamental justification for applying methods based on higher-order FSs to deal with everyday-life situations. Therefore, it comes as a natural consequence that such an extension should also be carried in the field of digital image processing.

THE IFIP FRAMEWORK

In quest for new theories treating imprecision, various higher-order extensions of FSs were proposed by different scholars. Among them, A-IFSs (Atanassov, 1986) provide a simple and flexible, yet solid, mathematical framework for coping with the intrinsic uncertainties characterizing real-world systems. A-IFSs are defined using two characteristic functions, namely the *membership* and the *non-membership* that do not necessarily sum up to unity. These functions assign to elements of the universe corresponding *degrees of belongingness* and *non-belongingness* with respect to a set. The membership and non-membership values induce an *indeterminacy index*, which models the *hesitancy* of deciding the degree to which an element satisfies a particular property. In fact, it is this additional degree of freedom that provides us with the ability to efficiently model and minimize the effects of uncertainty due to the imperfect and/or imprecise nature of information.

Hesitancy in images originates out of various factors, which in their majority are caused by inherent weaknesses of the acquisition and the imaging mechanisms. Distortions occurred as a result of the limitations of the acquisition chain, such as the quantization noise, the suppression of the dynamic range, or the nonlinear behavior of the mapping system, affect our certainty regarding the "*brightness*" or "*edginess*" of a pixel and therefore introduce a degree of hesitancy associated with the corresponding pixel. Moreover, dealing with "*qualitative*" rather than "*quantitativ*e" properties of images is one

of the sound advantages of fuzzy-based techniques. Qualitative properties describe in a more natural and human-centric manner image attributes, such as the "*contrast*" and the "*homogeneity*" of an image region, or the "*edginess*" of a boundary. However, as already pointed out, these terms are themselves imprecise and thus they additionally increase the uncertainty of image pixels. It is therefore a necessity, rather than a luxury, to employ A-IFSs theory to cope with the uncertainty present in real-world images.

In order to apply the IFIP framework, images should first be expressed in terms of elements of A-IFSs theory. Analyzing and synthesizing digital images to and from their corresponding intuitionistic fuzzy components is not a trivial task and can be carried out using either heuristic or analytic approaches.

Heuristic Modelling

As already stated, the factors introducing hesitancy in real-world images can be traced back to the acquisition stage of imaging systems and involve pixel degradation, mainly triggered by the presence of quantization noise generated by the A/D converters, as well as the suppression of the dynamic range caused by the imaging sensor. A main effect of quantization noise in images is that there exist a number of gray levels with zero, or almost zero, frequency of occurrence, while gray levels in their vicinity possess high frequencies. This is due to the fact that a gray level g in a digital image can be either (g+1) or (g-1) without any appreciable change in the visual perception.

An intuitive and heuristic approach to the modelling of the aforementioned sources of uncertainty in the context of A-IFSs was proposed by Vlachos & Sergiadis (Vlachos & Sergiadis, 2005) (Vlachos & Sergiadis, 2007 d) for gray-scale images, while an extension to color images was presented in Vlachos & Sergiadis (Vlachos & Sergiadis, 2006). The underlying idea involves the application of the concept of the *fuzzy histogram* of an image, which models the notion of the gray level "*approximately g*". The fuzzy histogram takes into account the frequency of neighboring gray levels to assess the frequency of occurrence of the gray level under consideration. Consequently, a quantitative measure of the quantization noise can be calculated as the normalized absolute difference between the ordinary (crisp) and fuzzy histograms.

Finally, to further incorporate the additional distortion factors into the calculation of hesitancy, parameters are employed that model the influence of the dynamic range suppression and the fact that lower gray levels are more prone to noise than higher ones.

Analytic Modelling

The analytic approach offers a more generic treatment to hesitancy modelling of digital images, since it does not require an a priori knowledge of the system characteristics, nor a particular pre-defined image acquisition model. Generally, it consists of sequential operations that primarily aim to optimally transfer the image from the pixel domain (PD) to the intuitionistic fuzzy domain (IFD), where the appropriate actions will be performed, using the fuzzy domain (FD) as an intermediate step. After the modification of the membership and non-membership components of the image in the IFD, an inverse procedure is carried out

Figure 1. Overview of the analytic IFIP framework
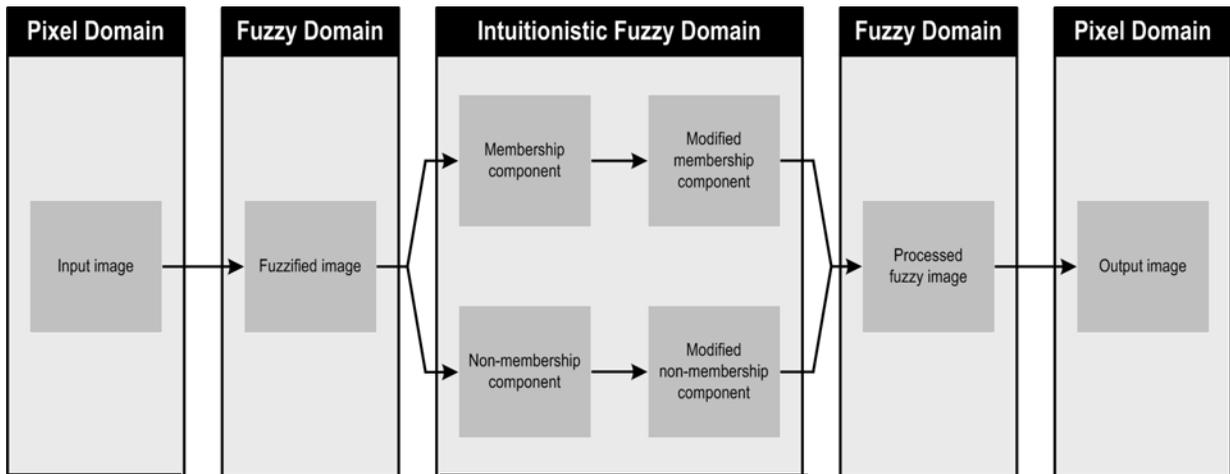
Figure 2. The process of fuzzification (from image properties to membership functions)
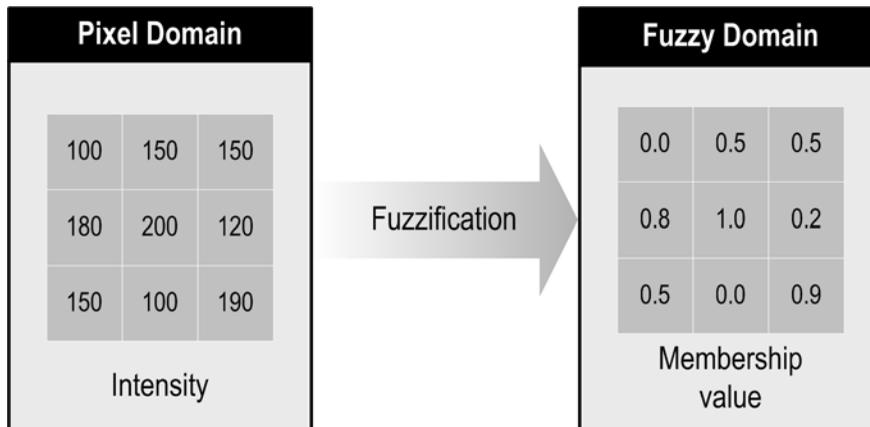
for transferring the image back to the PD. A block diagram illustrating the analytic modelling is given in Figure 1. Details on each of the aforementioned stages of IFIP are provided below.

Fuzzification

It constitutes the first stage of the IFIP framework, which assigns degrees of membership to image pixels with respect to an image property, such as "*brightness*", "*homogeneity*", or "*edginess*". These properties are application dependent and also determine the operations to be carried out in the following stages of the IFIP framework. For the task of contrast enhancement one may consider the "*brightness*" of gray levels and construct the corresponding FS "*Bright pixel*" or "*Dark pixel*" using different schemes that range from simple intensity normalization to more complex approaches involving knowledge extracted from a group of human experts (Figure 2).

Intuitionistic Fuzzification

Intuitionistic fuzzification is one of the most important stages of the IFIP architecture, since it involves the construction of the A-IFS that represents the image properties in the IFD. The analytic approach allows for an automated modelling of the hesitancy carried by image pixels, by rendering image properties directly from the FS obtained in the fuzzification stage through the use of *intuitionistic fuzzy generators* (Bustince, Kacprzyk & Mohedano, 2001). In order to construct an A-IFS that efficiently models a particular image property, tunable parametric intuitionistic fuzzy generators are utilized.

The underlying statistics of images are closely related to and soundly affect the process of hesitancy modelling. Different parameter values of the intuitionistic fuzzy generators produce different A-IFSs and therefore alternative representations of the image in the IFD are possible. Consequently, an optimization criterion should be employed, in order to select the parameter set that derives the A-IFS that optimally models the hesitancy of pixels from the multitude of possible representations. Such a criterion, that also encapsulates the image statistics, is the intuitionistic fuzzy entropy (Burillo & Bustince, 1996) (Szmidt & Kacprzyk, 2001) of the image under consideration. Therefore, the set of parameters that produce the A-IFS with the maximum

intuitionistic fuzzy entropy is considered as optimal. We refer to this process of selection as the *maximum intuitionistic fuzzy entropy principle* (Vlachos & Sergiadis, 2007 d). The optimal parameter set is then used to construct membership and non-membership functions corresponding to the intuitionistic fuzzy components of the image in the IFD. This procedure is schematically illustrated in Figure 3.

Modification of Intuitionistic Fuzzy Components

It involves the actual processing of the intuitionistic fuzzy components of the image with respect to a particular property. Depending on the desired image task one is about to perform, suitable intuitionistic fuzzy operators are applied to both membership and non-membership functions.

Intuitionistic Defuzzification

After obtaining the modified intuitionistic fuzzy components of the image, it is required that these components should be combined to produce the processed image in the FD. This procedure involves the embedding of hesitancy into the membership function. To carry out this task, we utilize suitable parametric intuitionistic fuzzy operators that de-construct an A-IFS into an FS. It should be stressed out that the final result soundly depends on the selected parameters of the aforementioned operators. Therefore, optimization criteria, such as the *maximization of the index of fuzziness* of the image, are employed to select the overall optimal parameters with respect to the considered image operation.

Defuzzification

The final stage of the IFIP framework involves the transfer of the processed fuzzy image into the PD. Depending on the desired image operation, various functions may be applied to carry out this task.

Applications

The IFIP architecture has been successfully applied to many image processing problems. Vlachos & Sergiadis (2007 d) exploited the potential of the framework in order to perform contrast enhancement to low-con-

trasted images. Different approaches were introduced, namely the intuitionistic fuzzy contrast intensification and the intuitionistic fuzzy histogram hyperbolization (IFHH). An extension of the IFHH technique to color images was proposed in Vlachos & Sergiadis (Vlachos & Sergiadis, 2007 b). Additionally, the effects of employing different intuitionistic fuzzification and intuitionistic defuzzification schemes to the performance of contrast enhancement algorithms was thoroughly studied and investigated in Vlachos & Sergiadis (2007)

(2007 d) and (2006 b), respectively. Application of A-IFSs theory to edge detection was also demonstrated in Vlachos & Sergiadis (Vlachos & Sergiadis, 2007 d), based on intuitionistic fuzzy similarity measures. The problem of image thresholding and segmentation under the context of IFIP, was also addressed (Vlachos & Sergiadis, 2006 a) using novel intuitionistic fuzzy information measures. Under the general framework of IFIP, the notions of the intuitionistic fuzzy histograms of a digital image were introduced (Vlachos

Figure 3. The process of intuitionistic fuzzification
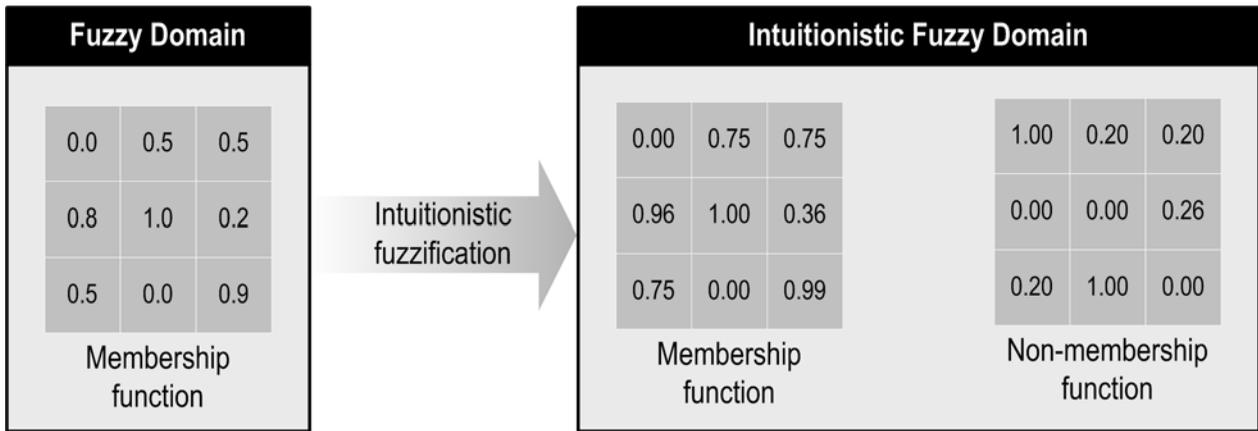
Figure 4. The stages of the IFIP framework for contrast enhancement
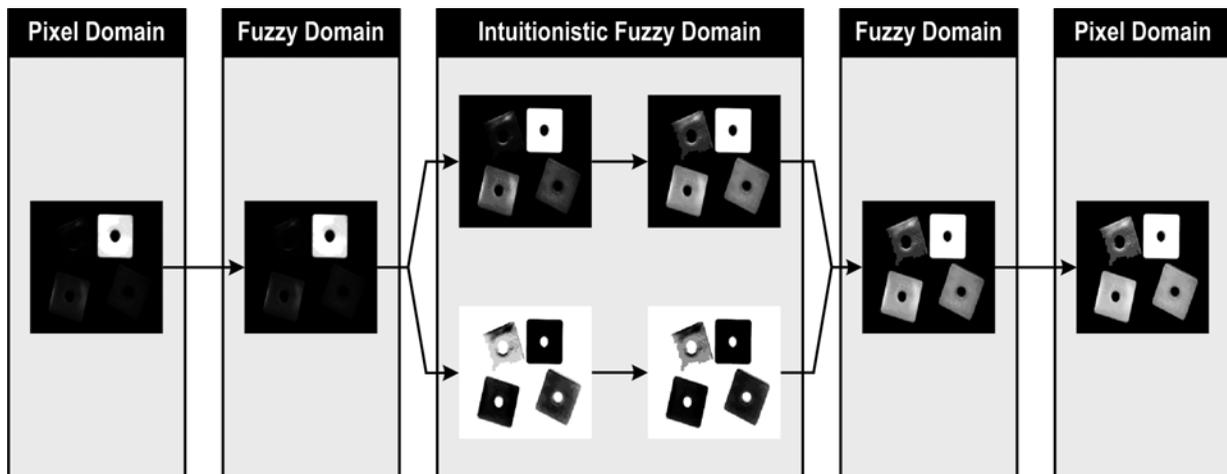
& Sergiadis, 2007 c) and their application to contrast enhancement was demonstrated (Vlachos & Sergiadis, 2007 a). Finally, the IFIP architecture was successfully applied in mammographic image processing (Vlachos & Sergiadis, 2007 d). Figure 4 illustrates the stages of IFIP in the case of the IFHH approach.

FUTURE TRENDS

Even though higher-order FSs have been widely applied to decision-making and pattern recognition problems, it seems that their application in the field of digital image processing is just beginning to develop. As a newly-introduced approach, the IFIP architecture remains a suggestively and challenging open field for future research. Therefore, it is expected that the IFIP framework will attract the interest of theoreticians and practitioners in the near future.

The proposed IFIP context bases its efficiency in the ability of A-IFSs to capture and render the hesitancy associated with image properties. Consequently, the analysis and synthesis of images in terms of elements of A-IFSs theory plays a key role in the performance of the framework itself. Therefore, the stages of intuitionistic fuzzification and defuzzification need to be further studied from an application point of view, to provide meaningful ways of extracting and embedding hesitancy from and to images. Finally, the IFIP architecture should be extended to image processing task handled today by FS theory, in order to investigate and evaluate its advantages and particular merits.

CONCLUSION

This article describes an intuitionistic fuzzy architecture for the processing of digital images. The IFIP framework exploits the potential of A-IFSs to efficiently model the uncertainties associated with image pixels, as well as with the definitions of their properties. The proposed methodology provides alternative approaches for analyzing/synthesizing images to/from their intuitionistic fuzzy components. Application of the IFIP framework to diverse imaging domains demonstrates its efficiency compared to traditional image processing techniques. It is expected that the proposed context will provide theoretician and practitioners with an alternative and

challenging way to perceive and deal with real-world image processing problems.

REFERENCES

Atanassov, K.T. (1986). Intuitionistic Fuzzy Sets. *Fuzzy Sets and Systems*, 20 (1), 87-96.

Atanassov, K.T., Koshelev, M., Kreinovich, V., Rachamreddy, B., & Yasemis, H. (1995). Fundamental Justification of Intuitionistic Fuzzy Logic and of Interval-Valued Fuzzy Methods. *Notes on Intuitionistic Fuzzy Sets*, 4 (2), 42-46.

Bezdek, J.C., Keller, J., Krisnapuram, R., & Pal, N.R. (1999). *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Springer.

Burillo, P., & Bustince, H. (1996). Entropy on Intuitionistic Fuzzy Sets and on Interval-Valued Fuzzy Sets. *Fuzzy Sets and Systems*, 78 (3), 305-316.

Bustince, H., Kacprzyk, J., & Mohedano, V. (2000). Intuitionistic Fuzzy Generators: Application to Intuitionistic Fuzzy Complementation. *Fuzzy Sets and Systems*, 114 (3), 485-504.

Chi, Z., Yan, H., & Pham, T. (1996). *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*, World Scientific Publishing Company.

Mendel, J.M., & Bob John, R.I. (2002). Type-2 Fuzzy Sets Made Simple. *IEEE Transactions on Fuzzy Systems*, 10 (2), 117-127.

Pal, S.K., & King, R.A. (1980). Image Enhancement Using Fuzzy Set. *Electronics Letters*, 16 (10), 376-378.

Pal, S.K., & King, R.A. (1981). Image Enhancement Using Smoothing with Fuzzy Sets. *IEEE Transactions on Systems, Man, and Cybernetics*, 11 (7), 495-501.

Pal, S.K., & King, R.A. (1982). A Note on the Quantitative Measurement of Image Enhancement Through Fuzziness. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4 (2), 204-208.

Prewitt, J.M.S. (1970). Object Enhancement and Extraction. *Picture Processing and Psycho-Pictorics* (pp. 75-149), Lipkin, B.S. Rosenfeld, A. (Eds.), Academic Press, New York.

Szmidt, E., & Kacprzyk, J. (2001). Entropy for Intuitionistic Fuzzy Sets. *Fuzzy Sets and Systems*, 118 (3), 467-477.

Vlachos, I.K., & Sergiadis, G.D. (2005). Towards Intuitionistic Fuzzy Image Processing. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation* (CIMCA 2005), Vienna, Austria.

Vlachos, I.K., & Sergiadis, G.D. (2006). A Heuristic Approach to Intuitionistic Fuzzification of Color Images. *Proceedings of the 7th International FLINS Conference on Applied Artificial Intelligence* (FLINS 2006), Genova, Italy.

Vlachos, I.K., & Sergiadis, G.D. (2006 a). Intuitionistic Fuzzy Information—Applications to Pattern Recognition. *Pattern Recognition Letters*, 28 (2), 197-206.

Vlachos, I.K., & Sergiadis, G.D. (2006 b). On the Intuitionistic Defuzzification of Digital Images for Contrast Enhancement. *Proceedings of the 7th International FLINS Conference on Applied Artificial Intelligence* (FLINS 2006), Genova, Italy.

Vlachos, I.K., & Sergiadis, G.D. (2007). A Two-Dimensional Entropic Approach to Intuitionistic Fuzzy Contrast Enhancement. *Proceedings of the International Workshop on Fuzzy Logic and Applications* (WILF 2007), Genova, Italy.

Vlachos, I.K., & Sergiadis, G.D. (2007 a). Hesitancy Histogram Equalization. *Proceedings of the IEEE International Conference on Fuzzy Systems* (FUZZ-IEEE 2007), London, United Kingdom.

Vlachos, I.K., & Sergiadis, G.D. (2007 b). Intuitionistic Fuzzy Histogram Hyperbolization for Color Images. *Proceedings of the International Workshop on Fuzzy Logic and Applications* (WILF 2007), Genova, Italy.

Vlachos, I.K., & Sergiadis, G.D. (2007 c). Intuitionistic Fuzzy Histograms of an Image. *Proceedings of the World Congress of the International Fuzzy Systems Association* (IFSA 2007), Cancun, Mexico.

Vlachos, I.K., & Sergiadis, G.D. (2007 d). Intuitionistic Fuzzy Image Processing. *Soft Computing in Image Processing: Recent Advances* (pp. 385-416), Nachtegael, M. Van der Weken, D. Kerre, E.E. Philips, W. (Eds.), Series: Studies in Fuzziness and Soft Computing, 210, Springer.

Vlachos, I.K., & Sergiadis, G.D. (2007 e). The Role of Entropy in Intuitionistic Fuzzy Contrast Enhancement. *Proceedings of the World Congress of the International Fuzzy Systems Association* (IFSA 2007), Cancun, Mexico.

KEY TERMS

Crisp Set: A set defined using a characteristic function that assigns a value of either 0 or 1 to each element of the universe, thereby discriminating between members and non-members of the crisp set under consideration. In the context of fuzzy sets theory, we often refer to crisp sets as "classical" or "ordinary" sets.

Defuzzification: The inverse process of fuzzification. It refers to the transformation of fuzzy sets into crisp numbers.

Fuzzification: The process of transforming crisp values into grades of membership corresponding to fuzzy sets expressing linguistic terms.

Fuzzy Logic: Fuzzy logic is an extension of traditional Boolean logic. It is derived from fuzzy set theory and deals with concepts of partial truth and reasoning that is approximate rather than precise.

Fuzzy Set: A generalization of the definition of the classical set. A fuzzy set is characterized by a membership function, which maps the members of the universe into the unit interval, thus assigning to elements of the universe degrees of belongingness with respect to a set.

Image Processing: Image processing encompasses any form of information processing for which the input is an image and the output an image or a corresponding set of features.

Intuitionistic Fuzzy Index: Also referred to as "hesitancy margin" or "indeterminacy index". It represents the degree of indeterminacy regarding the assignment of an element of the universe to a particular set. It is calculated as the difference between unity and the sum of the corresponding membership and non-membership values.

Intuitionistic Fuzzy Set: An extension of the fuzzy set. It is defined using two characteristic functions,

the membership and the non-membership that do not necessarily sum up to unity. They attribute to each individual of the universe corresponding degrees of belongingness and non-belongingness with respect to the set under consideration.

Membership Function: The membership function of a fuzzy set is a generalization of the characteristic function of crisp sets. In fuzzy logic, it represents the degree of truth as an extension of valuation.

Non-Membership Function: In the context of Atanassov's intuitionistic fuzzy sets, it represents the degree to which an element of the universe does not belong to a set.

Knowledge Management Systems Procedural Development

Javier Andrade
University of A Coruña, Spain

Santiago Rodríguez
University of A Coruña, Spain

María Seoane
University of A Coruña, Spain

Sonia Suárez
University of A Coruña, Spain

INTRODUCTION

The success of the organisations is increasingly dependant on the knowledge they have, to the detriment of other traditionally decisive factors as the work or the capital (Tissen, 2000). This situation has led the organisations to pay special attention to this new intangible item, so numerous efforts are being done in order to conserve and institutionalise it.

The Knowledge Management (KM) is a recent discipline replying this increasing interest; however, and despite its importance, this discipline is currently in an immature stage, as none of the multiple existing proposals for the development of Knowledge Management Systems (KMS) achieve enough detail for perform such complex task.

In order to palliate the previous situation, this work presents a methodological framework for the explicit management of the knowledge. The study has a formal basis for achieving an increased level of detail, as all the conceptually elements needed for understanding and representing the knowledge of any domain are identified. The requested descriptive character is achieved when basing the process on these elements and, in this way, the development of the systems could be guided more effectively.

BACKGROUND

During the last years numerous methodological frameworks for the development of KMS have arisen, the most important of which are the ones of Junnarkar (1997), Wiig et al (1997), Daniel et al (1997), Holsapple and Joshi (1997), Liebowitz and Beckman (Liebowitz, 1998; Beckman, 1997), Stabb and Schnurr (1999), Tiwana (2000) and Maté et al (2002). Nevertheless, the existing proposals do not satisfy adequately the needs of the organisation knowledge (Rubenstein-Montano, 2001; Andrade, 2003) due to their immaturity, mainly based on the following aspects:

1. The research efforts have been mainly focused on the definition of a process for KMS development, ignoring instead the study of the object to be managed: the knowledge.
2. The definition of such process has eluded in most of the cases the human factor and it has been restricted only to the technological viewpoint of the KM.

The first aspect regards the necessary study of the knowledge as basis for the definition of the Corporate Memory structure; this study should identify (i) the type of knowledge that has to be included in that repository and (ii) their descriptive properties for the Corporate Memory to include all the features of the knowledge items that it stores. The definition of that structure would enable also the definition of a descriptive process for creating KMS by using the different characteristics and types of knowledge.

However, and despite the influence that the object to be managed has on the management process, only the Wiig (1997) proposal pays attention to its study. Such

proposal identifies a small set of descriptors that support the formalisation (making explicit) of the knowledge although, (i) its identification does not result from an exhaustive study and (ii) it does not enable a complete formalisation as it is solely restricted to some generic properties.

The second step suggests that the whole process for KMS development should consider the technological as well as the human vision. The first one is focused on how obtaining, storing and sharing the relevant knowledge that exists within an organisation, by creating the Corporate Memory and the computer support system. The second vision involves, not only the creation of a collaborative atmosphere within the organisation in order to achieve the involvement of the workers in the KM program, but also the tendency to share their knowledge and use the one already provided by other members.

Despite the previous fact, the vast majority of the analysed approaches are solely focused on the technological KM viewpoint, which jeopardises the success of a KMS (Andrade, 2003). In fact, among the previously mentioned proposals, only the Tiwana (2000) proposal explicitly considers the human viewpoint by including a specific phase for it.

As a result of both aspects, the current proposals are restricted to a set of generic guides for performing KM, which is quite different from the formal and detailed vision that is being demanded. In other words, the current approaches indicate *what to do* but *not how to do it* (prescriptive viewpoint against descriptive/procedural viewpoint). In this scenario the developers of this type of systems have to elaborate their own *ad hoc* approach, achieving results that only depend on the experience and the capabilities of the development team.

DEVELOPMENT FOR KNOWLEDGE MANAGEMENT SYSTEMS

This section presents a methodological framework for the explicit KM that solves the previously mentioned problems. A study of the object to be managed has been performed for obtaining a knowledge formalisation schema, i.e., for knowing the relevant knowledge items and the characteristics/properties that should be made explicit. Using the results achieved after this study a methodological framework for KMS creation has been defined. Both aspects are following discussed.

Proposed Formalisation Schema

The natural language is the language par excellence for sharing knowledge. Due to this, a good identification of all the necessary elements for conceptualising (understanding) the knowledge of any domain (and therefore those for whom the respective formalisation mechanisms must be provided) can be done from the analysis of the different grammatical categories of the natural language: nouns, adjectives, verbs, adverbs, locutions and other linguistic expressions. This study, whose detailed description and applications have been described in several works (Andrade, 2006; Andrade, 2008), reveals that all the identified conceptual elements can be put into the following knowledge levels according to their function within the domain:

- Static. It regards the structural or operative knowledge domain, meaning domain facts that are true and that can be used in some operations as concepts, properties, relationships and constraints.
- Dynamic. It is related to the performance of the domain, that is, functionality, action, process or control: inferences, calculations and step sequence. This level can be divided into two sublevels:

- Strategic. It includes what to do, when and in what order (i.e., step factorisation).
- Tactical. It specifies how and when obtaining new operative knowledge (i.e., the description of a given step).

Every one of these levels approaches a different fragment of the organisation knowledge, although they all are obviously interrelated; in fact, the strategic level controls the tactical one, as for every last level/elemental step (strategic knowledge) the interferences and calculi must be indicated (tactical knowledge). Also the level of the operative knowledge is controlled by the other two, as it specifies how, not only the bifurcation points or execution alternatives are decided (strategic knowledge), but also how interferences and calculi are done (tactical knowledge).

Therefore, a KMS must provide support to all these levels. As it can be observed at Table 1, the main formalisation schema has been divided, on one hand, into several individual schemas corresponding to each one of the identified knowledge levels and, on the other, into

Table 1. Components defined for every identified schema

Schemas		Components
Common		Catalogue of terms
Dynamic	Strategic	Catalogue of non terminal steps
		Catalogue of terminal steps
	Tactical	Catalogue of tactical steps
Static	Operative	Catalogue of concepts
		Catalogue of relationships
		Catalogue of properties

a common one for the three levels, providing the global vision of the organisation knowledge. Therefore, the knowledge formalisation involves a dynamic schema including the strategic and tactical individual schemas, a dynamic schema including an operative schema, and a common schema, for describing the common aspects regardless the level. Every individual schema is also constituted by some components.

The catalogue of terms is a common component for the schemas, providing synonyms and abbreviations for identifying every knowledge asset within the organisation. The strategic schema describes the functional splitting of every KMS operation and also each identified step. As the description varies when the step is terminal or not (elemental step), two different components are needed for including all the characteristics of this level. The approach–procedural or algorithmic, for instance–should be described with detail for every asset included into the catalogue of terminal steps. All this information is included at the catalogue of tactical steps. Lastly, the static schema is made up of the catalogue of concepts–including the identified concepts and their description–, the catalogue of relationships–describing the identified relationships and their meaning–and the catalogue of properties–referring the properties of the previously mentioned concepts and relationships–.

The detailed description of this study, together with the descriptors of every component, can be found in (Andrade, 2008).

PROPOSED METHODOLOGICAL FRAMEWORK

The proposed process, whose basic structure is shown in Figure 1, has been elaborated bearing in mind the problems detected at the KM discipline and already mentioned throughout the present work.

As it can be noticed in the previous figure, this process includes the following phases:

1. Setting-up of the KM commission: the direction defines a KM commission for tracking and performing the KM project.
2. Scope identification. The problem to be approached would be specified by means of determining on where the present cycle of the KM project must have a bearing. In order to achieve this, the framework proposes the use of the SWOT analysis (Strengths, Weaknesses, Opportunities, Tricks), together with the proposal of Zack (1999).
3. Knowledge acquisition, including:

3.1. Identification of knowledge domains. The knowledge needs regarding the approached subject area are determined by means of different meetings involving the development team, the KM committee and the people responsible of every operation to be performed.

3.2. Capture of the relevant knowledge. The obtaining of all the possible knowledge related with the

Figure 1. Structure of the proposed process
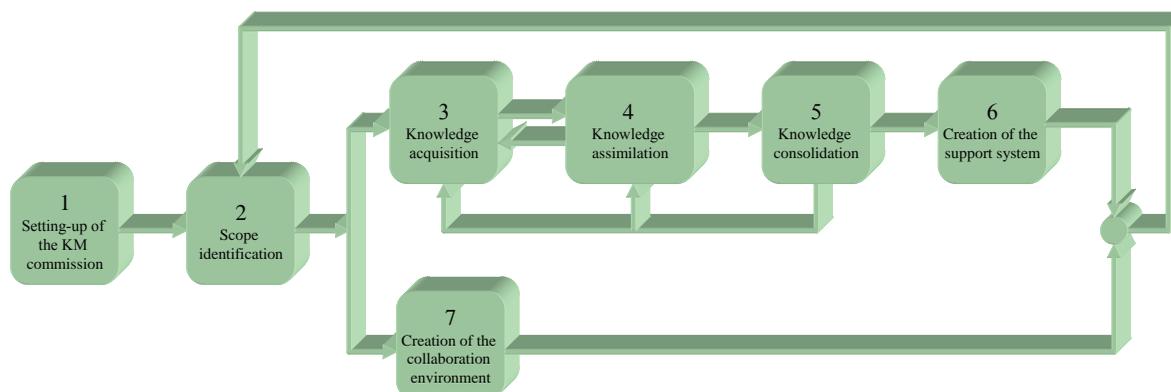
operation approached is based on the identified domains. This is done by means of:

(a) Identifying where the knowledge lies in. The KM commission is in charge of identifying and providing the human and non human knowledge sources that are going to be analysed.
(b) Determining the knowledge that has to be captured. As in the previous epigraph, it should be necessary to bear in mind the strategic, tactical and operative knowledge.
(c) Knowledge obtaining.

Obviously, when all the knowledge that is needed does not exist at the organisation it should be generated or imported.

4. Knowledge assimilation, comprising:

4.1. Knowledge conceptualisation. Its goal is the comprehension of the captured knowledge. It is recommended to start with the strategic knowledge for subsequently focusing on the tactical knowledge. As the strategic and tactical elements are understood, it would be necessary to assimilate arisen elements of the operative level.

4.2. Knowledge representation. The relevant knowledge has to be made explicit and formalised, according to the components (Andrade, 2008) summarised at Table 1. This in one of the main distinguishing

points of the proposal presented here, as the proposed formalisation schema indicates the specific descriptors needed for a correct and complete formalisation of the knowledge.

5. Knowledge consolidation, including:

5.1. Knowledge verification. In order to detect failures and omissions related with the represented knowledge it should be considered:

(a) Generic aspects. It has to be checked that any knowledge element (strategic, tactical and operative) is included into the catalogue of terms, that any term included there has been made explicit according to the type of knowledge and, that all the fields are completed.
(b) Strategic aspects. It should be verified that (i) any decision regarding an execution is made according to the existing operative knowledge, (ii) any last level step is associated to an existing tactical knowledge and, (iii) any non terminal step is correctly split. All the previous facts would be achieved by checking the accordance between the split tree and the content of the formalisation schema of the terminal strategic knowledge.
(c) Tactical aspects. It should be verified that: (i) the whole of the tactical knowledge is used in some of the last level steps of the strategic knowledge and that any operative knowledge related to the tactical knowledge is available. In order to achieve

this, the operative knowledge items will be represented as nodes within a knowledge map. This type of maps enable the graphic visualisation of how new elements are obtained from the existing ones. Once the map has been done, it should be scoured for checking that the whole of the operative knowledge has been included.

(d) Operative aspects. It should be confirmed that: (i) there are not isolated concepts, (ii) there are not attributes unrelated to a concept or to a relationship, (iii) there are not relationships associating non existing concepts or relationships and (iv) the whole of the operative knowledge is used in some of the tactical knowledge and/or in the decision making of the flow control of the strategic knowledge. In order to perform the three first verifications, a relationships diagram will be elaborated for graphically showing the existing relationships among the different elements of the operative knowledge. The syntax of this type of diagrams is analogous to the one of the class diagrams used in the methodologies of object-oriented software development. The verification of the last proposal will be done by using a knowledge map; the execution structures included into the content of the formalisation schema for the strategic knowledge of last level related to every process (the remaining inferior levels are included into the superior one) will be also used in this verification. With these two mentioned graphic representations it could be verified that every operative element is included into at least one of the representations.

5.2. Knowledge validation. In order to verify the knowledge represented and verified, the development team, the KM commission and the involved parts will revise:

(a) The knowledge splitting tree
(b) The knowledge map
(c) The relationship diagram
(d) The functional splitting tree
(e) The content of the formalisation schema

6. Creation of the support system, which is divided into:

6.1. Definition of the incorporation mechanisms. The KM commission and the development team determine the adequacy of the incorporation type (passive, active or their combination) according to criteria such as financial considerations or stored knowledge.

6.2. Definition of the notification mechanisms. The KM commission and the development team will establish the most suitable method for notifying the newly included knowledge. The notification can be passive or active; even the absence of notification could be considered.

6.3. Definition of the mechanisms for knowledge localisation. Several alternatives, such as the need of including intelligent searches or meta-searches, are evaluated.

6.4. Development of the KM support system. It will be necessary to define and to implement the corporate memory, the communication mechanisms and the applications for collaboration and team work.

6.5. Population of the corporate memory. Once the KM system has been developed. The knowledge captured, assimilated and consolidated will be included into the corporate memory.

7. Creation of the collaboration environment. The main goal of this phase is to promote and to improve the contribution of knowledge and its subsequent use by the organisation. It should be borne in mind the risk that involves the use of an unsuitable organisation culture or of inadequate tools for promotion and reward. The following strategies should be followed instead:

- Considering the employee worth according his/her knowledge contribution to the organisation
- Supporting and awarding the use of the organisational existing knowledge
- Promoting the relaxed dialogue among employees from different domains
- Promoting a good atmosphere among the employees
- Committing all the employees

FUTURE TRENDS

As it has been indicated, the KM discipline remains in an immature stage due to an inadequate viewpoint: the absence of a strict study for determining the relevant

knowledge and the characteristics that should be supported. Such situation has led to an important detail shortage of the existing proposals for KMS development, currently dependant solely from the individual good work of the developers.

The present proposal means a new viewpoint for developing this type of systems. However, it still remains a lot to do. As the authors are aware of the high grade of bureaucracy that might be needed for specifically following the present proposal, it should be expedited and characterised for specific domains. Nevertheless, this viewpoint could be considered as the key for achieving specific ontologies for KM in every domain.

CONCLUSION

This article has presented a methodological framework for the development of KMS that, differently from the existing proposals, is based on the strict study of the knowledge to be managed. This characteristic has provided the system with a higher procedural level of detail than the current proposals, as the elements conceptually needed for understanding and representing the knowledge of any domain have been identified and formalised.

REFERENCES

Andrade, J., Ares, J., García, R., Rodríguez, S., & Suárez, S. (2003). Lessons Learned for the Knowledge Management Systems Development. *Proc. 2003 IEEE International Conference on Information Reuse and Integration,* 471-477.

Andrade, J., Ares, J., García, R., Pazos, J., Rodríguez, S., & Silva S. (2006). Definition of a problem-sensitive conceptual modelling language: foundations and application to software engineering. *Information and Software Technology*, 48 (7), 517-531.

Andrade, J., Ares, J., García, R., Pazos, J., Rodríguez, S., & Silva S. (2008). Formal conceptualisation as a basis for a more procedural knowledge management. *Decision Support Systems, 45*(1), 164-179.

Beckman, T (1997).A Methodology for Knowledge Management. *International Association of Science and Technology for Development (IASTED) AI and Soft Computing Conference.*

Daniel, M., Decker, S., Domanetzki, A., Heimbrodt-Habermann, E., Höhn, F., Hoffmann, A., Röstel, H., Studer, R., & Wegner R. (1997). ERBUS-Towards a Knowledge Management System for Designers. *Proc. of the Knowledge Management Workshop at the 21st Annual German AI Conference.*

Holsapple, C., Joshi, K. (1997). Knowledge Management: A Three-Fold Framework. *Kentucky Initiative for Knowledge Management Paper*, n. 104.

Junnarkar, B. (1997). Leveraging Collective Intellect by Building Organizational Capabilities. *Expert Systems with Applications*, 13 (1), 29-40.

Liebowitz, J., & Beckman, T. (1998). *Knowledge Organizations. What Every Manager Should Know*. CRC Press.

Maté, J.L., Paradela, L.F., Pazos, J., Rodríguez-Patón, A. & Silva A. (2002). MEGICO: an Intelligent Knowledge Management Methodology. *Lecture Notes in Artificial Intelligence*, 2473, 102-107.

Staab, S., & Schnurr, H.P. (1999). Knowledge and Business Processes: Approaching and Integration. *Workshop on Knowledge Management and Organizational Methods. IJCAI99.*

Rubenstein-Montano, B., Liebowitz, J., Buchwalter, J., McCaw, D., Newman, B., & Rebeck K. (2001). A Systems Thinking Framework for Knowledge Management. *Decision Support Systems*, 31, 5-16.

Tissen, R., Andriessen, D., & Deprez, F. L. (2000). *The Knowledge Dividend*. Financial Times Prentice-Hall.

Tiwana, A. (2000). *The Knowledge Management Toolkit. Practical Techniques for Building a Knowledge Management System*. Prentice-Hall.

Wiig K., de Hoog, R., & van der Spek, R. (1997). Supporting Knowledge Management: a Selection of Methods and Techniques. *Expert Systems with Applications*, 13 (1), 15-27.

Zack, M. H. (1999). Developing a Knowledge Strategy. *California Management Review*, 41 (3), 125-145.

KEY TERMS

Commission of Knowledge Management: Team in charge of the Knowledge Management project.

Corporate Memory: Physical and persistent storage of the knowledge in an organisation. Its structure is determined by the knowledge formalisation schema.

Knowledge: Pragmatic level of information resulting from the combination of the information received with the individual experience.

Knowledge Formalisation Schema: Set of attributes for describing and formalising the knowledge.

Knowledge Management: Discipline that tries to suitably provide the adequate information and knowledge to the people indicated, whenever and how they need them. In such way these people will have all the necessary elements for best performing their tasks.

Knowledge Management System: System for managing knowledge in organizations, supporting the addition, storage, notification and localization of expertise and knowledge.

Methodological Framework: Approach for making explicit and structuring how a given task is performed.

K

Knowledge Management Tools and Their Desirable Characteristics

Juan Ares
University of A Coruña, Spain

Rafael García
University of A Coruña, Spain

María Seoane
University of A Coruña, Spain

Sonia Suárez
University of A Coruña, Spain

INTRODUCTION

The Knowledge Management (KM) is a recent discipline that was born under the idea of explicitly managing the whole existing knowledge of a given organisation (Wiig, 1995) (Wiig et al., 1997). More specifically, the KM involves providing the people concerned with the right information and knowledge at the most suitable level for them, when and how best suit them; in such way, these people will have all the necessary ingredients for choosing the best option when faced with a specific problem (Rodríguez, 2002).

As the knowledge, together with the ability for its best management, has turned into the key factor for the organizations to stand out, it is desirable to determine and develop the support instruments for the generation of such value within the organisations. This situation has been commonly accepted by several authors as (Brooking, 1996) (Davenport & Prusak, 2000) (Huang et al., 1999) (Liebowitz & Beckman, 1998) (Nonaka & Takeuchi, 1995) and (Wiig, 1993) among others. Technological tools should be available for diminishing the communication distance and for providing a common environment where the knowledge might accessible for being stored or shared.

As KM is a very recent discipline, there are few commercial software tools that deal with those aspects necessary for its approach. Most of the tools classified as KM-related are mere tools for managing documents, which is unsuitable for the correct management of the organisations knowledge. Bearing such problem in mind, the present work approaches the establishment of a KM support software tool based on the own definition of KM and on the existing tools. For achieving this, section 2 presents the market analysis that was performed for studying the existing KM tools, where not only their characteristics were analysed, but also the future needs of the knowledge workers. Following this study, the functionality that a KM support tool should have and the proposal for the best approach to that functionality were identified.

BACKGROUND

The first step for developing a complete KM support tool according to the present and future trade needs is the performance of a study of the existing market. After the initial identification of the characteristics that a KM support tool should have, a posterior work reveals how the studied tools provide support to every one of the previously identified characteristics. Lastly, an evaluation of the obtained results will be performed.

Characteristics to be Considered

The previously mentioned definition of KM was the basis for the identification of the characteristics to be considered, bearing in mind the different aspects that should be supported by the tool.

A KM tool should give support to the following aspects (Andrade et al., 2003a):

- Corporate Memory
- Yellow Pages
- Collaboration and Communication mechanisms

1. Corporate Memory
   The Corporate Memory compiles the knowledge that exists within an organisation for its workers disposal (Stein, 1995) (Van Heijst et al., 1997). Due to this, to compile and to make the relevant knowledge explicit is equally important than providing the suitable mechanisms for its correct and easy location, as well as recuperation.
2. Yellow Pages
   A KM program should not make the mistake of trying to capture and represent the whole existing knowledge of the organisation, as this attempt would not be feasible; in this sense, the relevant knowledge for the performance of the organisation should be the one to be included. However, not making all the knowledge explicit does not mean that it has to be obviated; for that reason, it is important to determine which knowledge has every individual at the organisation by means of the elaboration of the Yellow Pages. These ones identify and publish additional knowledge sources, human and non-human, that are at the organisation disposal (Davenport & Prusak, 2000).

3. Collaboration and Communication Mechanisms
   At the organisations the knowledge is share, as well as distributed, regardless of the automatism, or not, of the process. A knowledge transfer occurs every time that an employee asks a workmate of the adjoining office how to perform a given task. These daily knowledge transfers made the routine of the organisation up but, as they are local and fragmentary, some systems for user collaboration and communication should be therefore established. An adequate KM support tool should include mechanisms that guarantee the efficiency of the collaboration and the communication, regardless of the physical or temporal location of the interlocutors.

Analysed Tools

Once the aspects that a KM support tool should consider have been identified, the following step involves analysing how the current tools consider them.

With such purpose, the main so-named KM support tools that exist currently were analysed, discarding certain tools such as information search engines or simple applications for documents management, as they merely offer partial solutions.

Table 1. Tools analysed

	Corporate Memory	Yellow Pages	Collaboration and communication mechanisms
K-Factory	✓		✓
Norma K-Factor	✓	✓	✓
Hyperwave	✓	✓	✓
GTC	✓		✓
Epicentric	✓	✓	✓
Plumtree	✓	✓	✓
Intrasuite	✓	✓	✓
Coldata	✓	✓	✓
Intranets	✓	✓	✓
WebSpace	✓		✓
Knowledge Discovery System	✓	✓	✓
Documentum 5	✓		✓
Livelink (Opentext)	✓	✓	✓
Adenin	✓		✓

The analysis included thirteen tools (Table 1), all of them approaching at least two of the previously mentioned aspects. It should be highlighted that all the tools implement the Corporate Memory as a document warehouse, while the Yellow Pages appear as a telephone directory.

Results Evaluation

After the tools were analysed it was noticed that, for every aspect considered, there are some common elements. Bearing in mind these elements and the current needs, table 2 shows the desirable characteristics that a KM support tool should have.

The conclusions drawn after a deeper study on how the analysed tools approach the desirable characteristics are following presented.

Firstly it was observed that none of the tools classified as KM ones has the necessary structure for best identifying, formalising and sharing the relevant knowledge, as they solely perform documental management complemented, in the best of the cases, by some descriptive fields, the association to a contents tree or by means of links to another related documents. Such fact creates many problems, especially and due to the great data volume, the difficulty for selecting

the adequate knowledge that the user might need at a given moment. Therefore, and as it has been pointed previously, for the best use of the knowledge, it should be somehow structured. The communication supports are also quite important.

The characteristics of a KM support tool should be then necessarily defined, together with a guide for approaching them.

RECOMMENDED FEATURES

The approach to every one of the detected characteristics should be initiated as soon as the functionality that a support tool for the explicit management of the corporative knowledge might have been determined.

1. Corporate Memory: the organisation knowledge has to be physically stored somehow by means of a Corporate Memory for being adequately shared. A Corporate Memory is an explicit, independent and persistent knowledge representation (Stein, 1995) (Van Heijst et al., 1997) that can be considered as a knowledge repository from the individuals that work at a given organisation. The Corporate Memory should include the following aspects:

Table 2. Desirable characteristics of a KM support tool

Aspect	Desirable characteristic	
Corporate Memory		Knowledge formalisation
		Knowledge Incorporation
		New knowledge notification
		Search
Yellow Pages		Experts search
		Integration
Collaboration and communication mechanisms		Workgroup
		Workflow
		Management of time, tasks and resources
	Asynchronous communication	E-mail
		Forum
		Suggestion box
		Notice board
	Synchronous communication	Chat
		Electronic board
		Audio-conference
		Video-conference

1.1. Knowledge formalisation. Before being included into the Corporate Memory, the knowledge has to be formalised by means of the determination of, not only the relevant knowledge, but also the attributes that describe it. When performing this formalisation it should be born in mind that there are two types of knowledge; on one hand, the Corporate Memory must include the knowledge needed to describe the operations for performing an organisational task. On the other side, it is necessary to capture the knowledge that has been acquired by the individuals after their experience and life time. This markedly heuristic knowledge is known as *Learned Lessons*: positive as well as negative experiences that can be used for improving the future performance of the organisation (Van Heijst, 1997), and therefore refining its current knowledge.

a. Organisational knowledge (Andrade et al., 2003b). A KM system should consider different types of knowledge when structuring the relevant knowledge associated to the operations that exist at the organisation:

- Strategic or control knowledge: it indicates, not only what to do, but also why, where and when. For that reason, the constituents of the functional disintegration of every operation should be identified.

- Tactical: it specifies how and under what circumstances the tasks are done. This type of knowledge is associated with the execution process of every last-level strategic step.

b. Learned lessons. It is related to the experience and the knowledge that the individuals have with regards to their task. It provides the person who possesses it with the ability for refining both, the processes that follows at work and the already existing knowledge, in order to be more efficient. Whereas it's appropriate to create systems of learned lessons (Weber, 2001) in order to save this type of knowledge.

1.2. Incorporation mechanisms. The knowledge can be incorporated in an active or passive way (Andrade et al., 2003c). The active incorporation is based on the existence of a KM group in charge of looking after the quality of the knowledge that is going to be incorporated. This guarantees the quality of the knowledge included into the Corporate Memory but it also takes human resources up. Differently from the previous way, at the passive incorporation does not exist any group for quality evaluation, as the own individual ready to share knowledge and experience will be responsible for evaluating that the proposal fulfils the minimum requirements of quality and relevancy. The main advantage of the second alternative is that it does not take additional resources up. Bearing in mind the previous considerations, the active knowledge incorporation is preferred whenever it might be possible, as in such way the quality and the relevancy of the knowledge will be guaranteed.

1.3. Notification mechanisms. All the members of the organisation should be informed when a new knowledge is incorporated as this enables the refinement of their knowledge. The step previous to the notification is the definition of the group of people tan will be informed of the new appearance of a knowledge item. There are two alternatives (García et al., 2003): subscription, where every individual at the organisation might take out a subscription to certain preferred specific issues, and spreading, where the notification messages reach the workers without previous request. At the spreading, the messages can be sent to all the members of the organisation, but this is not advisable as the receptor would be not able of discern which ones of the vast amount of messages received might be interesting for him/her. Other spreading possibility would rely on an individual or a group that would be in charge of determining the addressees for every given message; this last option is quite convenient for the members of the organisation but it takes up a vast amount of resources that have to contain themselves

a lot of information regarding the interests of every one of the members.

1.4. Localisation mechanisms. The tool should be provided with some search mechanism in order to achieve the maximum possible profit from the captured and incorporated knowledge (Tiwana, 2000). It is necessary to reach an agreement between efficiency and functionality, as enough search options should be available without increasing the system complexity. For this reason, the following search mechanisms are suggested:

- Hierarchy search: this search catalogues the knowledge into a fixed hierarchy, in such way that the user might move through a group of links for refining the search performed.

- Attribute search: is based on the specification of terms in which the user is interested, resulting into some knowledge elements that might content those terms. This type of search provides more general results than the previous one.

2. Yellow Pages: a KM system should not try to capture and assimilate the whole of the knowledge that exists at the organisation as it would not be feasible. Therefore, the Yellow Pages are used for including, not only the systems that store knowledge, but also the individuals that have additional knowledge. Their elaboration is performed after determining the knowledge possessed by every individual at the organisation or by any other non human agents.

3. Collaboration and communication mechanisms: at the organisations, the knowledge is shared and distributed regardless the process might be automated or not. The technology helps the interchange of knowledge and ideas among the members of the organisation, as it enables bringing the best possible knowledge within reach of the individual who requires it. The collaboration and communication mechanisms detected are the following:

3.1 Asynchronous communication. Does not require the connection between the ends of the communication at the same time.

- E-mail. The electronic messenger enables the interchange of text and/or any other type of document among two or several users

- Forum. It consists of a Web page where the participants leave questions that do not have to be answered at that very moment. Other participants leave the answers which, together with the questions, can be seen by anyone entering the forum at any moment.

- Suggestion box. It enables sending suggestions or comments of any relevant aspect of the organisation to the adequate person or department.

- Notice board. It is a common space where the members of the organisation can publish some announcements appropriate for the public interest.

3.2 Synchronous communication. This type of interactive technology is based on real-time communications. Some of the most important systems are the following:

- Chat. It implies the communication among several people through the computer, as all the people connected can follow the communication, express an opinion, contribute ideas, make or answer questions when they decide.

- Electronic board. It provides the members of the organisation with a shared space for improving the interchange the ideas where everybody draws or writes.

- Audio conference. Two or more users can use real-time voice communication.

- Video conference. Two or more users can use real-time image communication.

FUTURE TRENDS

As it has been mentioned before, there is not a current KM tool that might cover adequately the organisational needs. This problem has been approached in the present work by trying to determine the functionality that any of these tools should incorporate. This is a first step that should be complemented with subsequent works, as it

K

is necessary to go deeper and determine better how to approach and implement the specified aspects.

CONCLUSION

The knowledge, either for its management or not, is transmitted within the organisations, although its existence does not imply its adequate use. There is a vast amount of knowledge where access is extremely difficult; this means that there are items from where no return is being achieved and that they are lost into the organisation. The KM represents the effort for capturing and getting benefits from the collective experience of the organisation by means of turning it accessible to any of its members. However, it could be stated that not a current tool is able to efficiently perform this task as, although there exist the so-named KM tools, they merely store documents and none of them performs the structuration of the relevant knowledge for its best use.

In order to palliate such problems, the present work proposes an approach based on a market research. It is as well based on the KM definition that indicates how to approach and defines the characteristics that a tool should have for working as facilitator of an adequate and explicit Knowledge Management.

REFERENCES

Andrade, J., Ares, J., García, R., Rodríguez, S., Silva, A., & Suárez, S. (2003a): Knowledge Management Systems Development: a Roadmap. Lecture Notes in Artificial Intelligence, 2775, 1008-1015.

Andrade, J.; Ares, J.; García, R.; Rodríguez, S. & Suárez, S. (2003b): Lessons Learned for the Knowledge Management Systems Development. In Proceedings of the 2003 IEEE International Conference on Information Reuse and Integration. Las Vegas (USA).

Brooking, A. (1996): Intellectual Capital. Core Asset for the Third Millennium Enterprise. International Thomson Business Press. London (UK).

Davenport, T. H. & Prusak, L. (2000): Working Knowledge: How Organizations Manage What They Know. Harvard Business School Press. Boston (USA).

García, R.; Rodríguez, S.; Seoane, M. & Suárez, S. (2003): Approach to the Development of Knowledge Management Systems. In Proceedings of the 10th. International Congress on Computer Science Research. Morelos (Mexico).

Huang, K. T.; Lee, Y. W. & Wang, R. Y. (1999): Quality Information and Knowledge. Prentice-Hall PTR. New Jersey (USA).

Liebowitz, J. & Beckman, T. (1998): Knowledge Organizations. What Every Manager Should Know. CRC Press. Florida (USA).

Nonaka, I. & Takeuchi, H. (1995): The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press. New York (USA).

Rodríguez, S. (2002): Un marco metodológico para la Knowledge Management y su aplicación a la Ingeniería de Requisitos Orientada a Perspectivas. PhD. Dissertation. School of Computer Science. University of A Coruña (Spain).

Stein, E. W. (1995): Organizational Memory: Review of Concepts and Recommendations for Management. International Journal of Information Management. Vol. 15. No. 2. PP: 17-32.

Tiwana, A. (2000): The Knowledge Management Toolkit. Prentice Hall.

Van Heijst, G.; Van der Spek, R. & Kruizinga, E. (1997): Corporate Memories as a Tool for Knowledge Management. Expert Systems with Applications. Vol. 13. No. 1. PP: 41-54.

Weber R., Aha, D. W., & Becerra-Fernandez, I. (2001): Intelligent Lessons Learned Systems. International Journal of Expert Systems Research and Applications, Vol. 20 No.1, PP: 17-34.

Wiig, K.; de Hoog, R. & Van der Spek, R. (1997): Supporting Knowledge Management: A Selection of Methods and Techniques. Expert Systems with Applications. Vol. 13. No. 1. PP: 15-27.

Wiig, K. (1993): Knowledge Management Foundations: Thinking about thinking. How People and Organizations Create, Represent and Use Knowledge. Schema Press. Texas (USA).

Wiig, K. (1995): Knowledge Management Methods: Practical Approaches to Managing Knowledge. Schema Press, LTD. Texas (USA).

KEY TERMS

Communication & Collaboration Tool: Systems that enable collaboration and communication among members of an organisation (i.e. chat applications, whiteboards).

Document Management: It is the computerised management of electronic, as well as paper-based documents.

Institutional Memory: It is the physical storage of the knowledge entered in an organization.

Knowledge: Pragmatic level of information that provides the capability of dealing with a problem or making a decision.

Knowledge Management: Discipline that intends to provide, at its most suitable level, the accurate information and knowledge for the right people, whenever they may needed and at their best convenience.

Knowledge Management Tool: Organisational system that connects people with the information and communication technologies, with the purpose of improving the share and distribution processes of the organisational knowledge.

Lesson Learned: Specific experience, positive or negative, of a certain domain. It is obtained into a practical context and can be used during future activities of similar contexts.

Yellow Page: It storages information about a human or non-human source that has additional and/or specialized knowledge about a particular subject.

Knowledge–Based Systems

Adrian A. Hopgood
De Montfort University, UK

INTRODUCTION

The tools of artificial intelligence (AI) can be divided into two broad types: knowledge-based systems (KBSs) and computational intelligence (CI). KBSs use explicit representations of knowledge in the form of words and symbols. This explicit representation makes the knowledge more easily read and understood by a human than the numerically derived implicit models in computational intelligence.

KBSs include techniques such as rule-based, model-based, and case-based reasoning. They were among the first forms of investigation into AI and remain a major theme. Early research focused on specialist applications in areas such as chemistry, medicine, and computer hardware. These early successes generated great optimism in AI, but more broad-based representations of human intelligence have remained difficult to achieve (Hopgood, 2003; Hopgood, 2005).

BACKGROUND

The principal difference between a knowledge-based system and a conventional program lies in its structure. In a conventional program, domain knowledge is intimately intertwined with software for controlling the application of that knowledge. In a knowledge-based system, the two roles are explicitly separated. In the simplest case there are two modules: the knowledge module is called the knowledge base and the control module is called the inference engine. Some interface capabilities are also required for a practical system, as shown in Figure 1.

Within the knowledge base, the programmer expresses information about the problem to be solved. Often this information is declarative, i.e. the programmer states some facts, rules, or relationships without having to be concerned with the detail of how and when that information should be applied. These latter details are determined by the inference engine, which uses the knowledge base as a conventional program uses a data file. A KBS is analogous to the human brain, whose control processes are approximately unchanging in their nature, like the inference engine, even though individual behavior is continually modified by new knowledge and experience, like updating the knowledge base.

As the knowledge is represented explicitly in the knowledge base, rather than implicitly within the structure of a program, it can be entered and updated with relative ease by domain experts who may not have any programming expertise. A knowledge engineer is someone who provides a bridge between the domain

Figure 1. The main components of a knowledge-based system
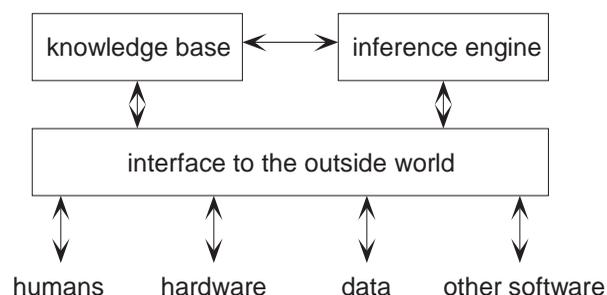
expertise and the computer implementation. The knowledge engineer may make use of meta-knowledge, i.e. knowledge about knowledge, to ensure an efficient implementation.

Traditional knowledge engineering is based on models of human concepts. However, it has recently been argued that animals and pre-linguistic children operate effectively in a complex world without necessarily using concepts. Moss (2007) has demonstrated that agents using non-conceptual reasoning can outperform stimulus–response agents in a grid-world test bed. These results may justify the building of non-conceptual models before moving on to conceptual ones.

TYPES OF KNOWLEDGE-BASED SYSTEM

Expert Systems

Expert systems are a type of knowledge-based system designed to embody expertise in a particular specialized domain such as diagnosing faulty equipment (Yanga, 2005). An expert system is intended to act like a human expert who can be consulted on a range of problems within his or her domain of expertise. Typically, the user of an expert system will enter into a dialogue in which he or she describes the problem – such as the symptoms of a fault – and the expert system offers advice, suggestions, or recommendations. It is often proposed that an expert system must offer certain capabilities that mirror those of a human consultant. In particular, it is often stated that an expert system must be capable of justifying its current line of inquiry and explaining its reasoning in arriving at a conclusion. This functionality can be integrated into the inference engine (Figure 1).

Rule-Based Systems

Rules are one of the most straightforward means of representing knowledge in a KBS. The simplest type of rule is called a production rule and takes the form:

```
if <condition> then <conclusion>
```

An example production rule concerning a boiler system might be:

```
/* rule1 */
if valve is open and flow is high then steam is escaping
```

Part of the attraction of using production rules is that they can often be written in a form that closely resembles natural language, as opposed to a computer language. The facts in a KBS for boiler monitoring might include:

```
/* fact1 */
valve is open
```

```
/* fact2 */
flow is high
```

One or more given facts may satisfy the condition of a rule, resulting in the generation of a new fact, known as a derived fact. For example, by applying rule1 to fact1 and fact2, fact3 can be derived:

```
/* fact3 */
steam is escaping
```

The derived fact may satisfy the condition of another rule, such as:

```
/* rule2 */
if steam is escaping or valve is stuck then outlet is blocked
```

This, in turn, may lead to the generation of a new derived fact or an action. Rule1 and rule2 are interdependent, since the conclusion of one can satisfy the condition of the other. The inter-dependencies amongst the rules define a network, as shown in Figure 2, known as an inference network.

It is the job of the inference engine to traverse the inference network to reach a conclusion. Two important types of inference engine can be distinguished: forward-chaining and backward-chaining, also known as data-driven and goal-driven, respectively. A KBS working in data-driven mode takes the available information, i.e. the given facts, and generates as many derived facts as it can. In goal-driven mode, evidence is sought to support a particular goal or proposition.

The data-driven (forward chaining) approach might typically be used for problems of interpretation, where the aim is to find out whatever the system can infer about some data. The goal-driven (backward chaining)

K

Figure 2. An inference network for a boiler system
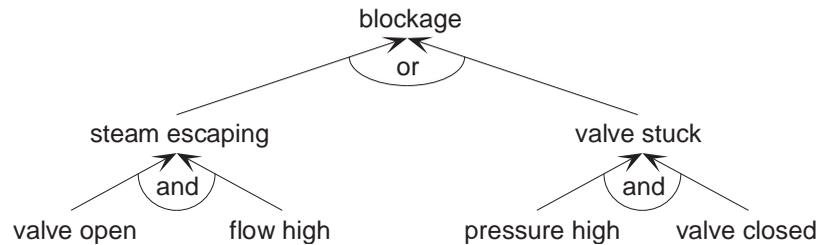
approach is appropriate when a more tightly focused solution is required, such as the generation of a plan for a particular goal. In the example of a boiler monitoring system, forward chaining would lead to the reporting of any recognised problems. In contrast, backward chaining might be used to diagnose a specific mode of failure by linking a logical sequence of inferences, disregarding unrelated observations.

The rules that make up the inference network in Figure 2 are used to link cause and effect:

if <cause> then <effect>

Using the inference network, an inference can be drawn that if the valve is open and the flow rate is high (the causes) then steam is escaping (the effect). This is the process of *deduction*. Many problems, such as diagnosis, involve reasoning in the reverse direction, i.e. the user wants to ascertain a cause, given an effect. This is *abduction*. Given the observation that steam is escaping, abduction can be used to infer that valve is open and the flow rate is high. However, this is only a valid conclusion if the inference network shows *all* of the circumstances in which steam may escape. This is the closed-world assumption.

If many examples of cause and effect are available, the rule (or inference network) that links them can be inferred. For instance, if every boiler blockage ever seen was accompanied by steam escaping and a stuck valve, then rule2 above might be inferred from those examples. Inferring a rule from a set of example cases of cause and effect is termed *induction*.

Hopgood (2001) summarizes deduction, abduction, and induction as follows:

- deduction: cause + rule $\Rightarrow$ effect
- abduction: effect + rule $\Rightarrow$ cause
- induction: cause + effect $\Rightarrow$ rule

Logic Programming

Logic programming describes the use of logic to establish the truth, or otherwise, of a proposition. It is, therefore, an underlying principle for rule-based systems. Although various forms of logic programming have been explored, the most commonly used one is the Prolog language (Bramer, 2005), which embodies the features of backward chaining, pattern matching, and list manipulation.

The Prolog language can be programmed declaratively, although an appreciation of the procedural behavior of the language is needed in order to program it effectively. Prolog is suited to symbolic problems, particularly logical problems involving relationships between items. It is also suitable for tasks that involve data lookup and retrieval, as pattern-matching is fundamental to the functionality of the language.

Symbolic Computation

A knowledge base may contain a mixture of numbers, letters, words, punctuation, and complete sentences. These symbols need to be recognised and processed by the inference engine. Lists are a particularly useful

data structure for symbolic computation, and they are integral to the AI languages Lisp and Prolog. Lists allow words, numbers, and symbols to be combined in a wide variety of ways. A list in the Prolog language might look like this:

[animal, [cat, dog], vegetable, mineral]

where this example includes a nested list, i.e. a list within a list. In order to process lists or similar structures, the technique of pattern matching is used. For example, the above list in Prolog could match to the list

[animal, [_, X], vegetable, Y]

where the variables X and Y would be assigned values of dog and mineral respectively. This pattern matching capability is the basis of an inference engine's ability to process rules, facts and evolving knowledge.

Uncertainty

The examples considered so far have all dealt with unambiguous facts and rules, leading to clear conclusions. In real life, the situation can be complicated by three forms of uncertainty:

Uncertainty in the Rule Itself

For example, rule1 (above) stated that an open valve and high flow rate lead to an escape of steam. However, if the boiler has entered an unforeseen mode, it made be that these conditions do not lead to an escape of steam. The rule ought really to state that an open valve and high flow rate will *probably* lead to an escape of steam.

Uncertainty in the Evidence

There are two possible reasons why the evidence upon which the rule is based may be uncertain. First, the evidence may come from a source that is not totally reliable. For example, in rule1 there may be an element of doubt whether the flow rate is high, as this information relies upon a meter of unspecified reliability. Second, the evidence itself may have been derived by a rule whose conclusion was probable rather than certain.

Use of Vague Language

Rule1, above, is based around the notion of a "high" flow rate. There is uncertainty over whether "high" means a flow rate of the order of $1cm^3s^{-1}$ or $1m^3s^{-1}$.

Two popular techniques for handling the first two sources of uncertainty are Bayesian updating and certainty theory (Hopgood, 2001). Bayesian updating has a rigorous derivation based upon probability theory, but its underlying assumptions, e.g., the statistical independence of multiple pieces of evidence, may not be true in practical situations. Certainty theory does not have a rigorous mathematical basis, but has been devised as a practical and pragmatic way of overcoming some of the limitations of Bayesian updating. It was first used in the classic MYCIN system for diagnosing infectious diseases (Buchanan, 1984). Other approaches are reviewed in (Hopgood, 2001), where it is also proposed that a practical non-mathematical approach is to treat rule conclusions as hypotheses that can be confirmed or refuted by the actions of other rules. Possibility theory, or fuzzy logic, allows the third form of uncertainty, i.e. vague language, to be used in a precise manner.

Decision Support and Analysis

Decision support and analysis (DSA) and decision support systems (DSSs) describe a broad category of systems that involve generating alternatives and selecting among them. Web-based DSA, which uses external information sources, is becoming increasingly important. Decision support systems that use artificial intelligence techniques are sometimes referred to as intelligent DSSs.

One clearly identifiable family of intelligent DSS is expert systems, described above. An expert system may contain a mixture of simple rules based on experience and observation, known as heuristic or shallow rules, and more fundamental or deep rules. For example, an expert system for diagnosing car breakdowns may contain a heuristic that suggests checking the battery if the car will not start. In contrast, the expert system might also contain deep rules, such as Kirchoff's laws, which apply to any electrical circuit and could be used in association with other rules and observations to diagnose any electrical circuit. Heuristics can often

provide a useful shortcut to a solution, but lack the adaptability of deep knowledge.

Building and maintaining a reliable set of cause–effect pairs in the form of rules can be a huge task. The principle of model-based reasoning (MBR) is that, rather than storing a huge collection of symptom–cause pairs in the form of rules, these pairs can be *generated* by applying underlying principles to the model. The model may describe any kind of system, including systems that are physical (Fenton, 2001), software-based (Mateis, 2000), medical (Montani, 2003), legal (Bruninghaus, 2003), and behavioral (De Koning, 2000). Models of physical systems are made up of fundamental components such as tubes, wires, batteries, and valves. As each of these components performs a fairly simple role, it also has a simple failure mode. Given a model of how these components operate and interact to form a device, faults can be diagnosed by determining the effects of local malfunctions on the overall device.

Case-based reasoning (CBR) also has a major role in DSA. A characteristic of human intelligence is the ability to recall previous experience whenever a similar problem arises. This is the essence of case-based reasoning (CBR), in which new problems are solved by adapting previous solutions to old problems (Bergmann, 2003).

Consider the example of diagnosing a broken-down car. If an expert system has made a successful diagnosis of the breakdown, given a set of symptoms, it can file away this information for future use. If the expert system is subsequently presented with details of another broken-down car of exactly the same type, displaying exactly the same symptoms in exactly the same circumstances, then the diagnosis can be completed simply by recalling the previous solution. However, a full description of the symptoms and the environment would need to be very detailed, and it is unlikely to be reproduced exactly. What is needed is the ability to identify a previous case, the solution of which can be reused or modified to reflect the slightly altered circumstances, and then saved for future use. Such an approach is a good model of human reasoning. Indeed case-based reasoning is often used in a semi-automated manner, where a human can intervene at any stage in the cycle.

FUTURE TRENDS

While large corporate knowledge-based systems remain important, small embedded intelligent systems have also started to appear in the home and workplace. Examples include washing machines that incorporate knowledge-based control and wizards for personal computer management. By being embedded in their environment, such systems are less reliant on human data input than traditional expert systems, and often make decisions entirely based on sensor data.

If AI is to become more widely situated into everyday environments, it needs to become smaller, cheaper, and more reliable. The next key stage in the development of AI is likely to be a move towards *embedded* AI, i.e. intelligent systems that are embedded in machines, devices, and appliances. The work of Choy (2003) is significant in this respect, as it demonstrates that the DARBS blackboard system can be ported to a compact platform of parallel low-cost processors.

In addition to being distributed in their applications, intelligent systems are also becoming distributed in their method of implementation. Complex problems can be divided into subtasks that can be allocated to specialized collaborative agents, bringing together the best features of knowledge-based and computation intelligence approaches (Li, 2003). As the collaborating agents need not necessarily reside on the same computer, an intelligent system can be both distributed and hybridized (Choy, 2004). Paradoxically, there is also a sense in which intelligent systems are becoming more integrated, as software agents share access to a single definitive copy of data or knowledge, accessible via the web.

CONCLUSION

As with any technique, knowledge-based systems are not suitable for all types of problems. Each problem calls for the most appropriate tool, but knowledge-based systems can be used for many problems that would be impracticable by other means. They have been particularly successful in narrow specialist domains. Building an intelligent system that can make sensible decisions about unfamiliar situations in everyday, non-specialist domains remains a severe challenge.

This development will require progress in simulating behaviors that humans take for granted – specifically perception, recognition, language, common sense, and adaptability. To build an intelligent system that spans the breadth of human capabilities is likely to require a hybrid approach using a combination of artificial intelligence techniques.

REFERENCES

Bergmann, R., Althoff, K.-D., Breen, S., Göker, M., Manago, M., Traphöner, R., and Wess, S. (2003). Developing Industrial Case-Based Reasoning Applications – the INRECA Methodology (2nd Edition). Lecture Notes in Artificial Intelligence, Vol. 1612. Springer - Buchreihe.

Bramer, M.A. (2005), Logic Programming with Prolog. Springer-Verlag, London.

Bruninghaus, S. and Ashley, K. D. (2003). Combining case-based and model-based reasoning for predicting the outcome of legal cases. Lecture Notes in Artificial Intelligence, 2689, 65-79.

Buchanan, B. G. and Shortliffe, E. H. (1984). Rule-Based Expert Systems: the MYCIN experiments of the Stanford Heuristic Programming Project, Addison-Wesley.

Choy, K.W., Hopgood, A.A., Nolle, L. and O'Neill, B.C. (2003). Design and implementation of an inter-process communication model for an embedded distributed processing network. International Conference on Software Engineering Research and Practice (SERP'03), Las Vegas, 239-245.

Choy, K.W., Hopgood, A.A., Nolle, L. and O'Neill, B.C. (2004). Implementation of a tileworld testbed on a distributed blackboard system. 18th European Simulation Multiconference (ESM2004), Magdeburg, Germany, 129-135.

De Koning, K., Bredeweg, B., Breuker, J., and Wielinga, B. (2000). Model-based reasoning about learner behaviour. Artificial Intelligence, 117, 173-229.

Fenton, W. G., Mcginnity, T. M., and Maguire, L. P. (2001). Fault diagnosis of electronic systems using intelligent techniques: a review. IEEE Transactions on Systems Man and Cybernetics Part C - Applications and Reviews, 31, 269-281.

Hopgood, A. A. (2001). Intelligent Systems for Engineers and Scientists, 2nd edition. CRC Press, Boca Raton.

Hopgood, A. A. (2003). Artificial intelligence: hype or reality? IEEE Computer, 6, 24-28.

Hopgood, A.A. (2005). The state of artificial intelligence. Advances in Computers, 65, 1-75.

Li, G., Hopgood, A.A. and Weller, M.J. (2003). Shifting Matrix Management: a model for multi-agent cooperation. Engineering Applications of Artificial Intelligence, 16, 191-201.

Mateis, C., Stumptner, M., and Wotawa, F. (2000). Locating bugs in Java programs - First results of the Java diagnosis experiments project. Lecture Notes in Artificial Intelligence, 1821, 174-183.

Montani, S., Magni, P., Bellazzi, R., Larizza, C., Roudsari, A. V., and Carson, E. R. (2003). Integrating model-based decision support in a multi-modal reasoning system for managing type 1 diabetic patients. Artificial Intelligence in Medicine, 29, 131-151.

Moss, N.G., Hopgood, A.A. and Weller, M.J. (2007). Can Agents without Concepts think? An Investigation using a Knowledge Based System. Proc. AI-2007: 27th SGAI International Conference on Artificial Intelligence, Cambridge, UK.

Yanga, B.S., Limb, D.S., and Tanc, A.C.C. (2005). VIBEX: an expert system for vibration fault diagnosis of rotating machinery using decision tree and decision table. Expert Systems with Applications, 28(4), 735-742.

KEY TERMS

Backward Chaining: Rules are applied through depth-first search of the rule base to establish a goal. If a line of reasoning fails, the inference engine must backtrack and search a new branch of the search tree. This process is repeated until the goal is established or all branches have been explored.

Case-Based Reasoning: Solving new problems by adapting solutions that were previously used to solve old problem.

Closed-World Assumption: The assumption that all knowledge about a domain is contained in the knowledge base. Anything that is not true according to the knowledge base is assumed to be false.

Deep Knowledge: Fundamental knowledge with general applicability, such as the laws of physics, which can be used in conjunction with other deep knowledge to link evidence and conclusions.

Forward Chaining: Rules are applied iteratively whenever their conditions are satisfied, subject to a selection mechanism known as conflict resolution when the conditions of multiple rules are satisfied.

Heuristic or Shallow Knowledge: Knowledge, usually in the form of a rule, that links evidence and conclusions in a limited domain. Heuristics are based on observation and experience, without an underlying derivation or understanding.

Inference Network: The linkages between a set of conditions and conclusions.

Knowledge-Based System: System in which the *knowledge base* is explicitly separated from the *inference engine* that applies the knowledge.

Model-Based Reasoning: The knowledge base comprises a model of the problem area, constructed from component parts. The inference engine reasons about the real world by exploring behaviors of the model.

Production Rule: A rule of the form if <condition> then <conclusion>.

K

Kohonen Maps and TS Algorithms

Marie-Thérèse Boyer-Xambeu
Université de Paris VII – LED, France

Ghislain Deleplace
Université de Paris VIII – LED, France

Patrice Gaubert
Université de Paris 12 – ERUDITE, France

Lucien Gillard
CNRS – LED, France

Madalina Olteanu
Université de Paris I – CES SAMOS, France

INTRODUCTION

In the analysis of a temporal process, Kohonen maps may be used together with time-series (TS) algorithms. Previous research aimed at combining Kohonen algorithms and Markov switching models in order to suggest a periodization of the international bimetallism in the 19th century (Boyer-Xambeu, Deleplace, Gaubert, Gillard and Olteanu, 2006). This research was based on an economic study of the international monetary system ruling at this time in Europe, which combined three monetary zones: a gold-standard one, centred in London, a bimetallic one, centred in Paris, and a silver-standard one, centred in Hamburg (Boyer-Xambeu, Deleplace and Gillard, 2006). The three major financial centres of that system (London, Paris, and Hamburg, hence the label LPH used hereafter) were linked through arbitrage operations between markets for gold and silver and markets for foreign exchange located in those centres. Since two metals, gold and silver, acted as monetary standards in that system, it worked as an international bimetallism. Its growing integration during half a century (from 1821 to 1873) was reflected in the convergence of the observed levels of the relative price of gold to silver in London, Paris, and Hamburg. However, this integration process was subject to various changes, which can be understood as exogenous shocks disturbing that process.

One such shock is vastly documented in the literature: the discovery of new gold mines in the United States and Australia, which led to a sudden decline in 1850 of the gold-silver price over all the markets in the world. This decline was not of the same magnitude everywhere, and therefore the spread between the London, Paris, and Hamburg gold-silver prices increased, stopping for a time the integration process of the system. This is what we will call a breaking in that process. The present paper aims at locating the major breakings occurring during the period of international bimetallism; a historical study could link them to special events, which operated as exogenous shocks on that system. The indicator of integration used is the spread between the highest and the lowest among the London, Paris, and Hamburg gold-silver prices.

Three algorithms are combined to study this integration: a periodization obtained with the SOM algorithm is confronted to the estimation of a two-regime Markov switching model, in order to give an interpretation of the changes of regime; at the same time change-points are identified over the whole period providing a more precise interpretation of these varying types of regulation.

Section 2 summarizes the results obtained with the SOM algorithm to differentiate the sub-periods obtained using the whole available data.

Section 3 presents the kind of model used and the results of its estimation using the new indicator, the

spread computed at each period of quotation between the three relative prices of gold in silver. The sub-periods are confronted to the two regimes obtained and some evidence of a relation between the regime and the volatility of the spread is presented.

Section 4 presents the technique used to identify change-points in the temporal process and some strong results of breaks in mean and in variance of the spread are obtained. They are interpreted in terms of monetary history as, for some of them, they are quite new in the literature of this domain.

Some further directions of research are indicated in conclusion.

THE SUB-PERIODS OBTAINED WITH A SOM ALGORITHM[1]

The Data

The relative prices of gold in silver are computed from the price of each metal observed, twice a week, in each of the three financial places, Paris, London and Hamburg (respectively, *poa*, *lgs*, and *hoa*), from the beginning of 1821 until the end of 1860. The same type of data is available for the exchange rates (Pound in Francs, Pound in Marks, Mark in Francs: respectively, *lpv*, *hlv*, and *phv*).

An observation is a set of twelve values, two quotations (Tuesday and Friday) for each of the six variables.

A computed variable has been added to emphasize the relation between the relative price of metals in Hamburg and the average level in Paris and London of this value (*hpl*).

Most of the time the quotations show rather small differences within a given week, but periods with important troubles, Paris in the late 1840s for instance, may be well separated from the more classical ones.

After the Kohonen classification using a grid of 25 nodes, a hierarchical ascending classification is used to produce a small number of macro classes, in this case 6 macro classes, corresponding to the main sub-periods. This latter classification is constructed with the code vectors obtained from the first process[2].

Characteristics of the Macro-Classes

Large sequences of contiguous weeks are grouped in the macro-classes, however a few years are fragmented in short periods situated in different classes

- Class 1 is constituted of 3 groups of years 1829-30, 1834-38, 1848-49 and a lot of fragments of other years
- Class 2 is more simple to describe with 3 intervals 1832-33, 1842-43 and 1846-47 and some sparse weeks from the 1830s.

They represent a central position contrasting to the well identified other classes:

- Class 3: 2 sets constituted of years 1824-25 and 1827-28, with almost no missing weeks in these intervals, indicating that this sub-period is very homogeneous
- Class 4: the end of year 1853 and the whole period 1854-60; again only a small number of weeks are missing for this continuous sub-period of more than seven years
- Class 5: 1821-24 and 1826-beginning 1827 plus small parts of 1830 and 1832
- Class 6: two sets 1839-41 and 1851-53

The means of the variables used to obtain the classification can be represented to illustrate the great differences appearing between the sub-periods. Changing hierarchies between the relative prices are the characteristic identifying the four last macro-classes.

Rearranging the various classes according to calendar time allows to distinguish between three sub-periods: a) the 1820s (classes 5 and 3, covering 1821 to 1828); b) the 1830s and 1840s (classes 1 and 2, covering 1829 to 1849); c) the 1850s (classes 6 and 4, covering 1851 to 1860).

Only the years 1839-41 resist to that rearrangement, since they belong to class 6, while they should appear in classes 1 and 2 relative to the 1830s and 1840s; some explanation will be suggested in the last section.

Fig. 1. exhibits two contrasted situations, where the gold-silver price is respectively low (class 4) and high (class 5) in all the three financial centres. Fig. 2. confirms that opposition, since the two classes are also

Figure 1. Gold-silver price and the 6 macro-classes
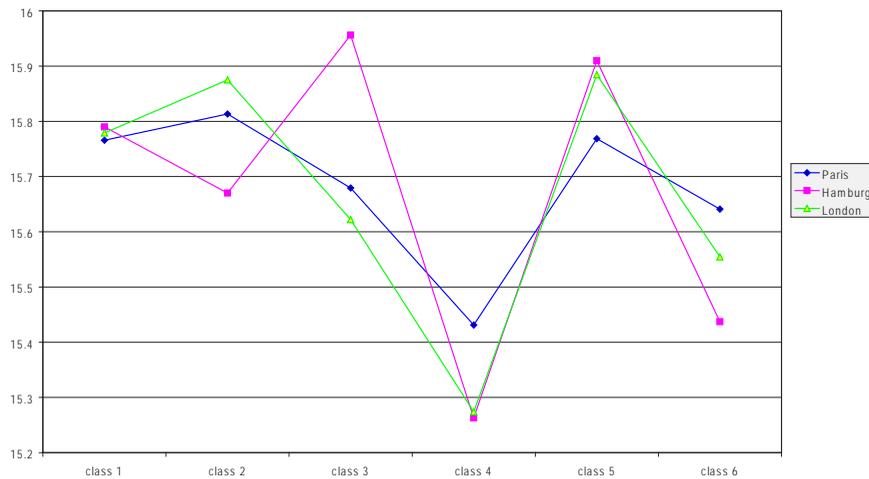
Figure 2. Exchange rates and the 6 macro-classes
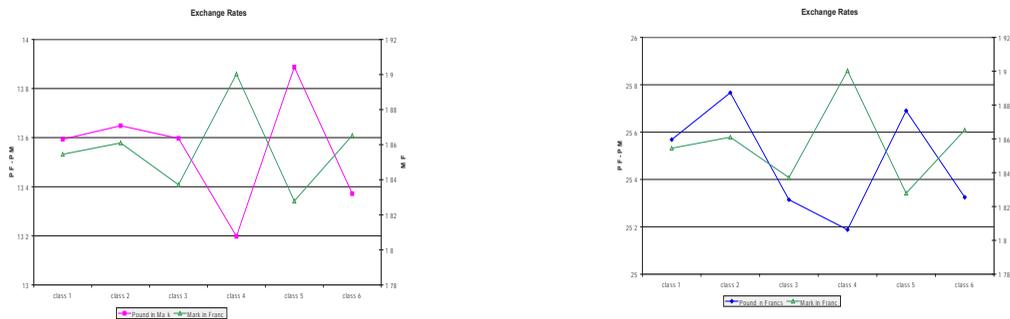
sharply contrasted by the levels of the exchange rates. Years 1821-23 and 1826 (class 5) are marked by a low mark/franc exchange rate and high gold-silver prices, the Hamburg one being higher than the Paris one; years 1854-60 (class 4) are marked by a high mark/franc exchange rate and low gold-silver prices, the Hamburg one being below the Paris one.

These remarks, which also apply respectively to the rest of the 1820s (class 3) and to the rest of the 1850s (class 6) are consistent with historical analysis: while the Hamburg mark was always anchored to silver, the French franc was during the 1820s and 1850s anchored to gold (in contrast with the 1830s and 1840s when it was anchored to silver); it is then normal that the mark

K

depreciated against the franc when silver depreciated against gold, and more in Hamburg than in Paris (as in class 5 and 3), and that the mark appreciated against the franc when silver appreciated against gold, and more in Hamburg than in Paris (as in class 4 and 6).

A MODEL FOR THE SPREAD BETWEEN THE HIGHEST AND THE LOWEST GOLD-SILVER PRICE

An Autoregressive Markov Switching Model

The key assumption is that the time series to be modeled follow a different pattern or a different model according to some unobserved, finite valued process. Usually, the unobserved process is a Markov chain whose states are called "regimes", while the observed series follows a linear autoregressive model whose coefficients depend on the current regime.

Let us put this in a mathematical language. Suppose that $(y_t)_{t \in Z}$ is the observed time series and that the unobserved process $(x_t)_{t \in Z}$ is a two-states Markov chain with transition matrix

$$A = \begin{pmatrix} p & 1-q \\ 1-p & q \end{pmatrix}, \text{ where } p, q \in]0,1[\qquad (1)$$

Then, assuming that y_t depends on the first l lags of time, we have the following equation of the model:

$$y_t = a_0^{x_t} + a_1^{x_t} y_{t-1} + \ldots + a_l^{x_t} y_{t-l} + \sigma^{x_t} \varepsilon_t \qquad (2)$$

where $a_i^{x_t} \in \{a_i^1, a_i^2\} \in R^2$ for every $i \in \{0,1,\ldots,l\}$, $\sigma^{x_t} \in \{\sigma^1, \sigma^2\} \in (R_+^*)^2$ and ε_t is a standard Gaussian noise.

The parameters of the model are then

$$\{a_0^1, a_1^1, \ldots, a_l^1, a_0^2, a_1^2, \ldots, a_l^2, \sigma^1, \sigma^2, p, q\}$$

and they are usually estimated by maximizing the log-likelihood function via an EM (Expectation – Maximization) algorithm[3].

Our characteristic of interest will be the "a posteriori" computed conditional probabilities of belonging to the first or to the second regime. Indeed, as our goal is to derive a periodization of the international bimetallism, the "a posteriori" computed states of the unobserved Markov chain will provide a natural one.

Although the results obtained with a switching Markov model are usually satisfying in terms of prediction and the periodizations are interesting and easily interpretable, a difficulty remains: how does one choose the number of regimes? In the absence of a complete theoretical answer, the criteria for selecting the "right" number of regimes are quite subjective from a statistical point of view[4].

The Results

In this paper we use a two-regime model to represent the spread computed with the gold-silver prices observed at each period on the three places. The transition matrix indicates good properties of stability:

$$\begin{pmatrix} 0.844298 & 0.253357 \\ 0.155702 & 0.746643 \end{pmatrix}$$

and no three regime model was found with an acceptable stability.

The first regime is a multilayer perceptron with one hidden layer, the second one is a simple linear model with one lag. Using the probabilities computed for each regime at each period, it may be interesting to study
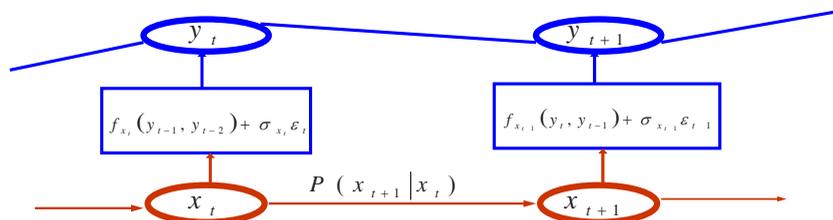
Table 1. Regime 1 and volatility of spread

Sub-periods	Number of obs.	% regime 1	Standard deviation of spread
1	483	0.733	0.053
2	335	0.627	0.061
3	191	0.445	0.075
4	376	0.816	0.044
5	303	0.625	0.050
6	390	0.723	0.049

the six sub-periods obtained and to observe the switch between the regimes along these periods of time.

Most of the time the regime 1 explains the spread (about 70% of the whole period) but important differences are to be noted between the sub-periods:

Classes 3 and 4 clearly contrast with, respectively, the highest and the lowest volatility of spread as they are ruled by, respectively, regime 2 and regime 1 models.

As will be explained later, further investigations have to be made with a more complex model and using a more adapted indicator of the arbitrages ruling the markets.

IDENTIFICATION OF CHANGE-POINTS: A GLOBAL VISION OF THE BIMETALLIST SYSTEM OF PAYMENTS

Elements About the Technique[5]

A different approach to model changes of regime in a time-series is to detect change-points or breaks. Here, the main assumption is that the whole series is observed and change-points are computed "a posteriori". Thus, this approach has not a predictive goal, but it is rather aimed at explaining the series by a piecewise stationary process which seems to be well adapted to our problem.

Mathematically, the model can be written as follows: let us consider the observed m-dimensional series $y_t = \{y_{1,t},...,y_{m,t}\}^T$, $t = 1,...,T$ and suppose that it is abruptly

changed. The changes, whose number and configuration are unknown, occur in the marginal distribution and may be in mean, in variance or in both mean and variance. We assume that there exists an integer K^* and a sequence of change-points $\tau^* = \{\tau_1^*,...,\tau_{K^*}^*\}$ with $\tau_0^* = 0 < \tau_1^* < ... < \tau_{K^*-1}^* < \tau_{K^*}^* = T$ such that $(\mu_k, \Sigma_k) \neq (\mu_{k+1}, \Sigma_{k+1})$ where $\mu_k = E(Y_t)$ and $\Sigma_k = Cov(Y_t) = E(Y_t - E(Y_t))(Y_t - E(Y_t))^T$, $\tau_{k-1}^* + 1 \leq t \leq \tau_k^*$.

The numbers of changes as well as their configuration are computed by minimizing a penalized contrast function. Details on the algorithms for computing the change-points configuration τ^* can be found in Lavielle and Teyssière (2006)[6].

Some Results and Interpretation

Applying this technique to the spread gave 7 change-points in mean and 4 in mean and variance.

Fig. 3 summarizes the spread, the four change-points (the first 4 green lines in chronological order) obtained in mean and variance, and the 2 last change-points in mean which correspond to a major break in the level of the gold-silver price, observed simultaneously on the three places and correspond to the great change in production of gold in United States.

A closer look at the spread between the highest and the lowest among the London, Hamburg and Paris gold-silver prices draws attention upon three episodes, each of them beginning with a break which sharply increases the spread and ends with another breaking which sharply narrows it (green vertical lines on Fig. 3). These episodes have in common to be linked to

Figure 3. Spread, change-points and probability of regime 1

SPREAD (Max-Min)
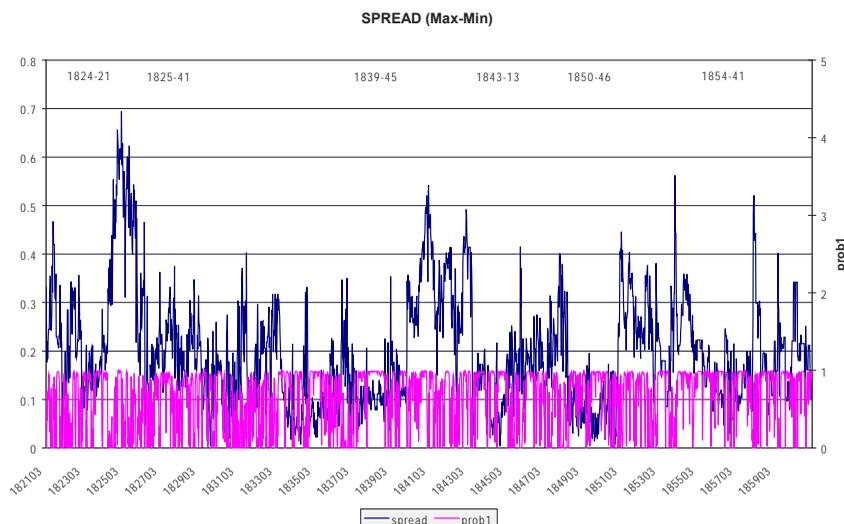
shocks affecting the integration process of the LPH system, although the shocks may have been asymmetrical (only one or two of the financial centres being initially hit) or symmetrical (the three of them being simultaneously hit).

The first episode runs from the 21st week of 1824 till the 41st week of 1825. The sharp initial increase in the spread may be explained by two opposite movements in London and Hamburg: on one side, heavy speculation in South-American bonds and Indian cotton fuelled in London the demand for foreign payments in silver, which resulted in a great increase in the price of silver and a corresponding decline in the gold-silver price; on the other side, the price of gold rose in Hamburg while the price of silver remained constant, sparkling the huge spread between the highest (Hamburg) and the lowest (London) gold-silver prices. More than one year later, the opposite movements took place: the price of gold plunged in Hamburg, while the price of silver remained at its height in London, under the influence of continuing speculation (which would end up in the famous banking crisis of December 1825); consequently the spread abruptly narrowed, this event being reflected by the breaking of the 41st week of 1825.

The second episode runs from the 45th week of 1839 till the 13th week of 1843. It started with the attempt of Prussia to unify the numerous German-speaking independent states in a common monetary zone, on a silver standard. Since the Bank of Hamburg maintained the price of silver fixed, that pressure on silver led to a drop in the Hamburg price of gold, and consequently in its gold-silver price, at a time when it was more or less stabilized in Paris. The spread between the highest (Paris) and the lowest (Hamburg) gold-silver price suddenly was enlarged, and during more than three years remained at a level significantly higher than during the 14 preceding years. This episode ended with the breaking of the 13th week of 1843, when, this shock having been absorbed, the gold-silver price in Hamburg went back in line with the price in the two other financial centres.

The third episode runs from the 46th week of 1850 till the 41st week of 1854. The shock was then symmetrical: London, Paris and Hamburg were hit by the pouring of gold following the discovery of Californian mines, and the sudden downward pressure on the world price of that metal. It took four years to absorb this

enormous shock, as reflected by the breaking of the 41st week of 1854.

CONCLUSION

In the three cases, the integration process of the LPH system, shown by the downward trend of the spread over half a century, was jeopardized by a shock: a speculative one in 1824, an institutional one in 1839, a technological one in 1850. But the effects of these shocks were absorbed after some time, thanks to active arbitrage operations between the three financial centres of the system. Generally, that arbitrage did not imply the barter of gold for silver but the coupling of a foreign exchange operation (on bills of exchange) with the transport of one metal only.

As a consequence, it would be appropriate in a further study to locate the breakings of another indicator of integration: the spread between a representative "national" gold-silver price and an arbitrated international gold-silver price taking into account the foreign exchange rates. At the same time it would be interesting to go further with the Markov switching model, trying more complete specifications.

REFERENCES

Boyer-Xambeu, M.-T., Deleplace, G. & Gillard, L. (1995), « Bimétallisme, taux de change et prix de l'or et de l'argent (1717-1873) », Economies et Sociétés 29, no. 7-8: 5-377.

Boyer-Xambeu, M.-T., Deleplace, G. & Gillard, L. (1997). 'Bimetallic Snake' and Monetary Regimes : The Stability of the Exchange Rate between London and Paris from 1796 to 1873. *Monetary Standards and Exchange Rates*, M.C. Marcuzzo, L.H. Officer, and A. Rosselli Eds., Routledge, London, 1997: 106-49.

Boyer-Xambeu, M.-T., Deleplace, G. & Gillard, L. (2006). International Bimetallism? Exchange Rates and Bullion Flows in Europe, 1821-1873, mimeo, Université Paris 8 – LED.

Boyer-Xambeu, M.-T., Deleplace, G., Gaubert, P., Gillard, L. & Olteanu, M. (2006). "Combining a Dynamic Version of Kohonen Algorithm and a Two-Regime Markov Switching Model: an Application to the Periodization of International Bimetallism (1821-1873)", Investigacion Operacional (forthcoming).

Cottrell, M., Fort, E.C. & Pagès, G. (1997), "Theoretical aspects of the Kohonen Algorithm" *WSOM'97*, Helsinki 1997.

Cottrell, M., Gaubert, P., Letremy, P., Rousset, P., "Analysing and Representing Multidimensional Quantitative and Qualitative Data. Demographic Study of the Rhone Valley. The Domestic Consumption of the Canadian Families", in *Kohonen Maps*, E. Oja and S. Kaski Eds., Elsevier Science, Amsterdam, 1999.

Hamilton, J. D. (1989). "A new approach to the economic analysis of non-stationary time series and the business cycle", Econometrica, 57, 357-84

Kohonen, T. *Self-Organization and Associative Memory*. (3rd edition 1989), Springer, Berlin, 1984.

Lavielle, M. (1999), "Detection of multiple changes in a sequence of dependant variables", Stochastic Process. Appl., vol. 83, pp 79-102.

Lavielle, M. & Teyssière, G. (2005), "Adaptive detection of multiple change-points in asset price volatility", in Teyssière G. & Kirman Eds. A. *Long-Memory in Economics*, Springer, Berlin, pp.129-156.

Lavielle, M. & Teyssière, G. (2006), "Detection of Multiple Change-Points in Multivariate Time Series", Lithuanian Mathematical Journal, vol. 46, n° 3, pp 287-306.

Maillet B., Olteanu M., Rynkiewicz J. (2004), "Non-linear Analysis of Shocks when Financial Markets are Subject to Changes in Regime", Proceedings of ESANN 2004, p. 87-92

Olteanu M., Rynkiewicz J. (2006), "Estimating the Number of Regimes in an Autoregressive Model with Markov Switching", IOR 2006, La Habana.

Rynkiewicz J. (2004), "Estimation of linear autoregressive models with Makov-switching", Investigacion Operacional, La Havane, Cuba. Vol. 25:2, p. 166-173

Teyssière, G. (2003), "Interaction models for common long-range dependence in asset price volatility", in Rangarajan G. and Ding M. Eds., *Processes with Long Range Correlations: Theory and Applications*, Lectures Notes in Physics, 621, Springer, Berlin, pp. 251-269.

KEY TERMS

Change-Point: Instant of time where the basic parameters of time series change (in mean and/or in variance); the series may be considered as a piecewise stationary process between two change-points

Gold-Silver Price: Ratio of the market price of gold to the market price of silver in one place. The stability of that ratio through time and the convergence of its levels in the various places constituting the international bimetallism (see that definition) are tests of the integration of that system.

International Arbitrage: Activity of traders in gold and silver and in foreign exchange, which consisted in comparing their prices in different places, and in moving the precious metals and the bills of exchange accordingly, in order to make a profit. Arbitrage and monetary rules were the two factors explaining the working of international bimetallism (see that definition).

International Bimetallism: An international monetary system (see that definition) which worked from 1821 to 1873. It was based on gold and silver acting as monetary standards, either together in the same country (like France) or separately in different countries (gold in England, silver in German and Northern states). The integration of that system was reflected in the stability and the convergence of the observed levels of the relative price of gold to silver (see that definition) in London, Paris, and Hamburg.

International Monetary System: A system linking the currencies of various countries, which ensures the stability of the exchange rates between them. Its working depends on the monetary rules adopted in each country and on international arbitrage (see that definition) between the foreign exchange markets. Historical examples are the gold-standard system (1873-1914) and the Bretton-Woods system (1944-1976). The paper studies some characteristics of another historical example: international bimetallism (see that definition).

Markov Switching Model: An autoregressive model where the process linking a present value to its lags is an hidden Markov chain defined by its transition matrix

SOM Algorithm: An unsupervised technique of classification (Kohonen,1984) combining adaptative learning and neighbourhood to construct a very stable classification, with a more simple interpretation ('Kohonen maps') than other techniques.

ENDNOTES

[1] Details may be found in Boyer-Xambeu, ..., Olteanu, 2006.
[2] See Cottrell M., Fort…(1997) and Cottrell M., Gaubert…(1999).
[3] See Rynkiewicz (2004) and Maillet et al. (2004).
[4] See Olteanu et al. (2006).
[5] The authors are very grateful to Gilles Teyssière for a significant help on this part.
[6] See also Lavielle M. and Teyssière G. (2005), Teyssière G. (2003) and Lavielle M. (1999).

K

Learning in Feed–Forward Artificial Neural Networks I

Lluís A. Belanche Muñoz
Universitat Politècnica de Catalunya, Spain

INTRODUCTION

The view of artificial neural networks as adaptive systems has lead to the development of ad-hoc generic procedures known as *learning rules*. The first of these is the Perceptron Rule (Rosenblatt, 1962), useful for single layer feed-forward networks and linearly separable problems. Its simplicity and beauty, and the existence of a *convergence theorem* made it a basic departure point in neural learning algorithms. This algorithm is a particular case of the Widrow-Hoff or *delta* rule (Widrow & Hoff, 1960), applicable to continuous networks with no hidden layers with an error function that is quadratic in the parameters.

BACKGROUND

The first truly useful algorithm for feed-forward multilayer networks is the *backpropagation* algorithm (Rumelhart, Hinton & Williams, 1986), reportedly proposed first by Werbos (1974) and Parker (1982). Many efforts have been devoted to enhance it in a number of ways, especially concerning speed and reliability of convergence (Haykin, 1994; Hecht-Nielsen, 1990). The backpropagation algorithm serves in general to compute the gradient vector in all the first-order methods, reviewed below.

Neural networks are trained by setting values for the network parameters $\underline{w}$ to minimize an error function $E(\underline{w})$. If this function is quadratic in $\underline{w}$, then the solution can be found by solving a linear system of equations (e.g. with Singular Value Decomposition (Press, Teukolsky, Vetterling & Flannery, 1992)) or iteratively with the delta rule. The minimization is realized by a variant of a *gradient descent* procedure, whose ultimate outcome is a local minimum: a $\underline{w}^*$ from which any infinitesimal change makes $E(\underline{w}^*)$ increase, that may not correspond to one of the global minima. Different solutions are found by starting at different initial states. The process is also perturbed by round-off errors. Given $E(\underline{w})$ to be minimized and an initial state $\underline{w}^0$, these methods perform for each iteration the updating step:

$$\underline{w}^{i+1} = \underline{w}^i + \alpha_i \underline{u}^i \qquad (1)$$

where $\underline{u}^i$ is the *minimization direction* (the direction in which to move) and $\alpha_i \in R$ is the *step size* (how far to make a move in $\underline{u}^i$), also known as the *learning rate* in earlier contexts. For convenience, define $\Delta\underline{w}^i = \underline{w}^{i+1} - \underline{w}^i$. Common stopping criteria are:

1. A maximum number of presentations of D (*epochs*) is reached.
2. A maximum amount of computing time has been exceeded.
3. The evaluation has been minimized below a certain tolerance.
4. The gradient norm has fallen below a certain tolerance.

LEARNING ALGORITHMS

Training algorithms may require information from the objective function only, the gradient vector of the objective function or the Hessian matrix of the objective function:

- *Zero-order* training algorithms make use of the objective function only. The most significant algorithms are *evolutionary algorithms*, which are global optimization methods (Goldberg, 1989).
- *First-order* training algorithms use the objective function and its gradient vector. Examples are *Gradient Descent*, *Conjugate Gradient* or *Quasi-Newton* methods, which are all local optimization methods (Luenberger, 1984).
- *Second-order* training algorithms make use of the objective function, its gradient vector and its Hessian matrix. Examples are *Newton's method*

and the *Levenberg-Marquardt algorithm*, which are local optimization methods (Luenberger, 1984).

First-order methods. The *gradient* $\nabla E\underline{w}$ of an *s*-dimensional function is the vector field of first derivatives of $E(\underline{w})$ w.r.t. $\underline{w}$,

$$\nabla E\underline{w} = (\frac{\partial E(\mathbf{w})}{\partial w_1}, ..., \frac{\partial E(\mathbf{w})}{\partial w_s}) \qquad (2)$$

Here $s = dim(\underline{w})$. A linear approximation to $E(\underline{w})$ in an infinitesimal neighbourhood of an arbitrary point $\underline{w}^i$ is given by:

$$E(\underline{w}) \approx E(\underline{w}^i) + \nabla E\underline{w}(\underline{w}^i) \cdot (\underline{w} - \underline{w}^i) \qquad (3)$$

We write $\nabla E\underline{w}(\underline{w}^i)$ for the gradient $\nabla E\underline{w}$ evaluated at $\underline{w}^i$. These are the first two terms of the Taylor expansion of $E(\underline{w})$ around $\underline{w}^i$. In *steepest* or *gradient* descent methods, this local gradient alone determines the minimization direction u^i. Since, at any point $\underline{w}^i$, the gradient $\nabla E\underline{w}(\underline{w}^i)$ points in the direction of fastest increase of $E(\underline{w})$, an adjustment of $\underline{w}^i$ in the negative direction of the local gradient leads to its maximum decrease. In consequence the direction $u^i = -\nabla E\underline{w}(\underline{w}^i)$ is taken.

In conventional steepest descent, the step size α_i is obtained by a *line search* in the direction of u^i: how far to go along u^i before a new direction is chosen. To this end, evaluations of $E(\underline{w})$ and its derivatives are made to locate some nearby local minimum. Line search is a move in the chosen direction u^i to find the minimum of $E(\underline{w})$ along it. For this one-dimensional problem, the simplest approach is to proceed along u^i in small steps, evaluating $E(\underline{w})$ at each sampled point, until it starts to increase. One often used method is a divide-and-conquer strategy, also called Brent's method (Fletcher, 1980):

1. Bracket the search by setting three points $a < b < c$ along u^i such that $E(au^i) > E(bu^i) < E(cu^i)$. Since E is continuous, there is a local minimum in the line joining a to c.
2. Fit a parabola (a quadratic polynomial) to a,b,c.
3. Compute the minimum μ of the parabola in the line joining a to c. This value is an approximation of the minimum of E in this interval.

4. Set three new points a,b,c out of μ and the two points among the old a,b,c having the lowest E. Repeat from 2.

Although it is possible to locate the nearby global minimum, the cost can become prohibitedly high. The line search can be replaced by a fixed step size α, which has to be carefully chosen. A sufficiently small α is required such that $\alpha \nabla E\underline{w}(u^i)$ is effectively very small and the expansion (3) can be applied. A too large value might cause to overshoot or lead to divergent oscillations and a complete breakout of the algorithm. On the other hand, very small values translate in a painfully slow minimization. In practice, a trial-and-error process is carried out.

A popular heuristic is a historic average of previous changes to exploit tendencies and add inertia to the descent, accomplished by adding a so-called *momentum* term $\beta_i \Delta \underline{w}^{i-1}$, where $\Delta \underline{w}^{i-1}$ is the previous weight update (Rumelhart, Hinton & Williams, 1986). This term helps to avoid or smooth out oscillations in the motion towards a minimum. In practice, it is set to a constant value $\beta \in (0.5, 1)$. Altogether, for steepest descent, the update equation (1) reads:

$$\underline{w}^{i+1} = \underline{w}^i + \alpha_i u^i + \beta \Delta \underline{w}^{i-1} \qquad (4)$$

where $u^i = -\nabla E\underline{w}(\underline{w}^i)$ and $\Delta \underline{w}^{i-1} = \underline{w}^i - \underline{w}^{i-1}$. This method is very sensitive to the chosen values for α_i and β, to the point that different values are required for different problems and even for different stages in the learning process (Toolenaere, 1990). The inefficiency of the steepest descent method stems from the fact that both u^i and α_i are somewhat poorly chosen. Unless the first step is chosen leading straight to a minimum, the iterative procedure is very likely to wander with many small steps in zig-zag. Therefore, these methods are quite out of use nowadays. A method in which both parameters are properly chosen is the *conjugate gradient*.

Conjugate Gradient. This minimization technique (explained at length in Shewchuck, 1994) is based on the idea that a new direction u^{i+1} should not spoil previous minimizations in the directions $u^i, \underline{u}^{i-1}, ..., \underline{u}^1$. This is the case if we simply choose $u^i = -g^i$, where $g^i = \nabla E\underline{w}(\underline{w}^i)$, as was found above for steepest descent. At most points on $E(\underline{w})$, the gradient does *not* point directly towards the minimum. After a line minimization, the new gradient g^{i+1} is orthogonal to the line

search direction, that is, $g^{i+1} \cdot u^i = 0$. Thus, successive search directions will also be orthogonal, and the error function minimization will proceed in zig-zag in a extremely slow advance to a minimum.

The solution to this problem lies in determining consecutive search directions u^{i+1} in such a way that the component of the gradient parallel to u^i (which has just been made to be zero, because we minimized in that direction) remains zero, so consecutive search directions complement each other, avoiding the possibility of spoiling the progress done in previous iterations.

Let us assume a line minimization has just been made along u^i, starting from the current weights $\underline{w}^i$; we have thus found a new point $\underline{w}^{i+1}$ for which

$$\nabla E\underline{w}(\underline{w}^{i+1}) \cdot u^i = 0 \qquad (5)$$

holds. The next search direction u^{i+1} is chosen to retain the property that the component of the gradient parallel to u^i, remains zero:

$$\nabla E\underline{w}(\underline{w}^{i+1} + \alpha_i u^{i+1}) \cdot u^i = 0 \qquad (6)$$

Expanding (6) to first order in α_i, and applying (5), we obtain the condition (Bishop, 1995):

$$u^{i+1} \cdot H_{\underline{w}}(\underline{w}^{i+1}) \cdot u^i = 0 \qquad (7)$$

If the error surface is quadratic, (7) holds regardless of the value of α_i, because the Hessian is constant, and higher-order terms in the previous expansion vanish. Search directions u^{i+1}, u^i fulfilling (7) are said to be *conjugate*. It can be proven that, in these conditions, it is possible to construct a sequence $\underline{u}^1, ..., \underline{u}^s$ such that $\underline{u}^s$ is conjugate to all previous directions, so that the minimum can be located in at most $s = dim(\underline{w})$ steps.

The *conjugate gradient* technique departs from (1) but sets $u^{i+1} = -g^{i+1} + \beta_i u^i$, setting $\underline{u}^1 = -g^1$. It turns out that the β_i can be found without explicit knowledge of the Hessian and the various versions of conjugate gradient are distinguished by the manner in which the parameter β_i is set. For the *Polak-Ribière* updating:

$$\beta_i = \frac{g^{i+1} \cdot (g^{i+1} - g^i)}{\underline{g}^i \underline{g}^i} \qquad (8)$$

This is the inner product of the previous change in the gradient with the current gradient, divided by the squared norm of the previous gradient. For the *Fletcher-Reeves* updating:

$$\beta_i = \frac{\underline{g}^{i+1} \cdot \underline{g}^{i+1}}{\underline{g}^i \underline{g}^i} \qquad (9)$$

This is the ratio of the squared norm of the current gradient to the squared norm of the previous gradient. The α_i can be found by performing a line search (at each iteration step) to determine the optimal distance to move along the current train direction; that is, a line minimization of $\nabla E\underline{w}(\underline{w}^i + \alpha_i u^i)$ w.r.t. α_i. This results in a particular case of steepest descent with momentum, where the parameters α_i, β_i are determined at each iteration. For a quadratic error surface $E(\underline{w})$, the method finds the minimum after at most $s = dim(\underline{w})$ steps, without calculating the Hessian. In practice $E(\underline{w})$ may be far from being quadratic so the technique needs to be run for many iterations and augmented with a criterion to reset the search vector to the negative gradient direction $u^{i+1} = -g^{i+1}$ after every s steps (Press, Teukolsky, Vetterling & Flannery, 1992). The *scaled* version (Møller, 1993) takes some account of the non-quadratic nature of the error function. With these enhancements the method is generally believed to be fast and reliable. Contrary to steepest descent, it is relatively insensitive to its parameters –the line search for α_i and the variants of computing β_i– if they are set within a reasonable tolerance. There is some evidence that the Polak-Ribière formula accomplishes the transition to further iterations more efficiently: when it runs out of steam, it tends to reset the train direction u^i to be down the local gradient, which is equivalent to beginning the conjugate-gradient procedure again.

The Newton training algorithm. First-order approximations ignore the curvature of $E(\underline{w})$. This can be fixed by considering the second-order term of the Taylor expansion around some $\underline{w}^i$ in weight space:

$$E(\underline{w}) \approx E(\underline{w}^i) + \nabla E\underline{w}(\underline{w}^i) \cdot (\underline{w} - \underline{w}^i) + 1/2 (\underline{w} - \underline{w}^i) \cdot H_{\underline{w}}(\underline{w}^i) \cdot (\underline{w} - \underline{w}^i) \qquad (10)$$

where $H_{\underline{w}} = \nabla\nabla E\underline{w}$ is the Hessian $s \times s$ matrix of second derivatives, with components

$$H_{\underline{w}} = (h_{ij}), \quad h_{ij} = \frac{\partial^2 E(\mathbf{w})}{\partial w_i \partial w_j}.$$

Again, $H_{\underline{w}}(\underline{w}^i)$ indicates the evaluation of $H_{\underline{w}}$ in $\underline{w}^i$. The Hessian traces the curvature of the error function in weight space, portraying information about how $\nabla E\underline{w}$ changes in different directions. Given a direction $\boldsymbol{u}^i$ from $\underline{w}^i$, the product $H_{\underline{w}}(\underline{w}^i)\cdot\boldsymbol{u}^i$ is the rate of change of the gradient along $\boldsymbol{u}^i$ from $\underline{w}^i$. Differentiating (10) w.r.t. $\underline{w}$, a local approximation of the gradient around $\underline{w}^i$ is:

$$\nabla E\underline{w} \approx \nabla E\underline{w}(\underline{w}^i) + H_{\underline{w}}(\underline{w}^i)\cdot(\underline{w}-\underline{w}^i) \tag{11}$$

For points close to $\underline{w}^i$, (10) and (11) give reasonable approximations to the error function $E(\underline{w})$ and to its gradient. Setting (11) to zero, and solving for $\underline{w}$, one gets:

$$\underline{w}^* = -H^{-1}{}_{\underline{w}}(\underline{w}^i)\cdot\nabla E\underline{w}(\underline{w}^i) \tag{12}$$

Newton's method uses the second partial derivatives of the objective function and hence is a second order method, finding the minimum of a quadratic function in just one iteration. The vector $H^{-1}{}_{\underline{w}}(\underline{w}^i)\cdot\nabla E\underline{w}(\underline{w}^i)$ is known as the Newton train direction. Since higher-order terms have been neglected, the update formula (12) is used iteratively to find the optimal solution. An exact evaluation of the Hessian is computationally demanding if done at each stage of an iterative algorithm; the Hessian matrix must also be inverted. Further, the Newton train direction may move towards a maximum or a saddle point rather than a minimum (if the Hessian is not positive definite) and the error would not be guaranteed to be reduced. This motivates the development of alternative approximation methods.

The quasi-Newton training algorithm. As shown above, the iterative formula used in Newton's method is

$$\underline{w}^{i+1} = \underline{w}^i - H^{-1}{}_{\underline{w}}(\underline{w}^i)\cdot\nabla E\underline{w}(\underline{w}^i) \tag{13}$$

The basic idea behind the quasi-Newton method is to approximate $H^{-1}{}_{\underline{w}}$ by another matrix G, using only the first partial derivatives of the error function. If $H^{-1}{}_{\underline{w}}$ is approximated by G, Equation (13) can be expressed as

$$\underline{w}^{i+1} = \underline{w}^i - \alpha^{*(i)}(G(\underline{w}^i)\cdot\nabla E\underline{w}(\underline{w}^i)) \tag{14}$$

where $\alpha^{*(i)}$ can be considered as the optimal train rate along the train direction $G(\underline{w}^i)\cdot\nabla E\underline{w}(\underline{w}^i)$. The gradient

descent direction method can be obtained by setting $G=I$.

In order to implement Equation (14), an approximate inverse G of the Hessian matrix is needed. Two commonly used algorithms are the Davidon-Fletcher-Powell (DFP) algorithm and the Broyden-Fletcher-Goldfarb-Shanno (BGFS) algorithm. The DFP algorithm is given by

$$G^{(i+1)} = G^{(i)} + \frac{(\underline{w}^{i+1} - \underline{w}^i)\otimes(\underline{w}^{i+1} - \underline{w}^i)}{(\underline{w}^{i+1} - \underline{w}^i).(\underline{g}^{i+1} - \underline{g}^i)}$$
$$+ \frac{[G^{(i)}.(\underline{g}^{i+1} - \underline{g}^i)]\otimes(G^{(i)}.[\underline{g}^{i+1} - \underline{g}^i])}{(\underline{g}^{i+1} - \underline{g}^i).G^{(i)}.(\underline{g}^{i+1} - \underline{g}^i)}, \tag{15}$$

where $\otimes$ denotes outer product, which is a matrix: the i,j component of $\boldsymbol{u}\otimes\boldsymbol{v}$ is $u_i v_j$. The BGFS algorithm is exactly the same, but with one additional term:

$$G^{(i+1)} = G^{(i)} + \frac{(\underline{w}^{i+1} - \underline{w}^i)\otimes(\underline{w}^{i+1} - \underline{w}^i)}{(\underline{w}^{i+1} - \underline{w}^i).(\underline{g}^{i+1} - \underline{g}^i)}$$
$$+ \frac{[G^{(i)}.(\underline{g}^{i+1} - \underline{g}^i)]\otimes(G^{(i)}.[\underline{g}^{i+1} - \underline{g}^i])}{(\underline{g}^{i+1} - \underline{g}^i).G^{(i)}.(\underline{g}^{i+1} - \underline{g}^i)},$$
$$+ ((\boldsymbol{g}^{i+1}-\boldsymbol{g}^i)\cdot G^{(i)}\cdot(\boldsymbol{g}^{i+1}-\boldsymbol{g}^i))\,\underline{v}\otimes\underline{v}, \tag{16}$$

where the vector $\underline{v}$ is given by

$$\underline{v} = \frac{(\underline{w}^{i+1} - \underline{w}^i)}{(\underline{w}^{i+1} - \underline{w}^i).(\underline{g}^{i+1} - \underline{g}^i)}$$
$$- \frac{G^{(i)}.(\underline{g}^{i+1} - \underline{g}^i)}{(\underline{g}^{i+1} - \underline{g}^i).G^{(i)}.(\underline{g}^{i+1} - \underline{g}^i)}. \tag{17}$$

It is generally recognized that the BFGS scheme is empirically superior to the DFP scheme (Press, Teukolsky, Vetterling & Flannery, 1992).

The Levenberg-Marquardt method, like the quasi-Newton methods, was designed to approach second-order training speed without computing the Hessian. When the objective function is a sum of squares (a very common case in neural networks), the Hessian matrix can be approximated as $H=J^T J$ and the gradient computed as $\boldsymbol{g}=J^T\underline{e}$, where J is the Jacobian matrix,

containing first derivatives of the network errors with respect to the weights, and $\underline{e}$ is the vector of network errors. The Jacobian can be computed via standard backpropagation, a much less complex process than computing the Hessian.

The Levenberg-Marquardt algorithm uses this approximation to the Hessian in the following Newton-like update:

$$\underline{w}^{i+1} = \underline{w}^i - (J^T J + \mu I)^{-1} J^T \underline{e} \qquad (18)$$

When the scalar μ is zero, this is Newton's method using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in objective function) and is increased only when a temptative step would increase it. When computation of the Jacobian becomes prohibitedly high for big networks on a large number of training examples, the quasi-Newton method is preferred.

THE BACK-PROPAGATION ALGORITHM

This algorithm represented a breakthrough in connectionist learning due to the possibility to compute the gradient of the error function $\nabla E_{\underline{w}}(\underline{w}^i)$ recursively and efficiently. It is so-called because the components of the gradient concerning weights belonging to output units are computed first, and propagated backwards (toward the inputs) to compute the rest, in the order marked by the layers. Intuitively, the algorithm finds the extent to which the adjustment of one connection will reduce the error in the training examples (the partial derivative of the error function $E(\underline{w})$ with respect to the connection) and therefore the algorithm computes the full gradient vector $\nabla E_{\underline{w}}$.

To derive the algorithm, we introduce some notation. Given $f: R^n \rightarrow R^m$ the function to be approximated, we depart from a finite training set D of p samples of f,

$$D = \{ <\boldsymbol{x}_1, \boldsymbol{y}_1>, ..., <\boldsymbol{x}_p, \boldsymbol{y}_p> \}, f(\boldsymbol{x}_\mu)_k + \varepsilon = y_{\mu,k} \qquad (19)$$

For simplicity, we assume that the loss is the square error and define:

$$E(\underline{w}) = 1/2 \sum_{\mu=1}^{p} \sum_{k=1}^{m} (y_{\mu,k} - \zeta^{\mu,c+1}{}_k)^2 \qquad (20)$$

where $\zeta^{\mu,c+1}{}_k = F_{\underline{w}}(\boldsymbol{x}_\mu)_k$ is the k-th component of the network's response to input pattern $\boldsymbol{x}_\mu$ (the network has $c+1$ layers, of which c are hidden). For a given input pattern $\boldsymbol{x}_\mu$, define:

$$E^\mu(\underline{w}) = 1/2 \sum_{k=1}^{m} (y_{\mu,k} - \zeta^{\mu,c+1}{}_k)^2 \qquad (21)$$

so that

$$E(\underline{w}) = \sum_{\mu=1}^{p} E^\mu(\underline{w}).$$

The computation of a single unit i in layer l, $1 \le l \le c+1$ upon presentation of pattern $\boldsymbol{x}_\mu$ to the network may be expressed $\varsigma_i^{\mu,l} = g(\hat{\varsigma}_i^{\mu,l})$, with g a smooth function –as the sigmoidals– and

$$\hat{\varsigma}_i^{\mu,l} = \sum_j w_{ij}^l \varsigma_j^{\mu,l-1}.$$

The first outputs are then defined $\zeta^{\mu,0}{}_i = x_{\mu,i}$. A single weight w_{ij}^l denotes the connection strength from neuron j in layer l-1 to neuron i in layer l, $1 \le l \le c+1$.

If the gradient-descent rule (1) with constant α is followed, then $\boldsymbol{u}^i = -\nabla E_{\underline{w}}(\underline{w}^i)$. Together with (2), the increment Δw_{ij} in a single weight w_{ij}^l of $\underline{w}$ is:

$$\Delta w_{ij}^l = -\alpha \frac{\partial E(\mathbf{w})}{\partial w_{ij}^l} = -\alpha \sum_{\mu=1}^{p} \frac{\partial E^\mu(\mathbf{w})}{\partial w_{ij}^l} = -\alpha \sum_{\mu=1}^{p} \Delta^\mu w_{ij}^l \qquad (22)$$

We have:

$$\Delta^\mu w_{ij}^l = \frac{\partial E^\mu(\mathbf{w})}{\partial w_{ij}^l} = \frac{\partial E^\mu(\mathbf{w})}{\partial \varsigma_i^{\mu,l}} \frac{\partial \varsigma_i^{\mu,l}}{\partial \hat{\varsigma}_i^{\mu,l}} \frac{\partial \hat{\varsigma}_i^{\mu,l}}{\partial w_{ij}^l} \qquad (23)$$

Proceeding from right to left in (23):

$$\frac{\partial \hat{\varsigma}_i^{\mu,l}}{\partial w_{ij}^l} = \varsigma_j^{\mu,l-1} \qquad (24)$$

$$\frac{\partial \varsigma_i^{\mu,l}}{\partial \hat{\varsigma}_i^{\mu,l}} = \frac{dg(\hat{\varsigma}_i^{\mu,l})}{d\hat{\varsigma}_i^{\mu,l}} = g'(\hat{\varsigma}_i^{\mu,l}) \tag{25}$$

Assuming g to be the logistic function with slope β:

$$g'(\hat{\varsigma}_i^{\mu,l}) = \beta\, g(\hat{\varsigma}_i^{\mu,l})\,(1-g(\hat{\varsigma}_i^{\mu,l})) = \beta\, \varsigma_i^{\mu,l}\,(1-\varsigma_i^{\mu,l}) \tag{26}$$

The remaining expression

$$\frac{\partial E^{\mu}(\mathbf{w})}{\partial \varsigma_i^{\mu,l}}$$

is more delicate and constitutes the core of the algorithm. We develop two separate cases: $l=c+1$ and $l<c+1$. The first case corresponds to output neurons, for which an expression of the derivative is immediate from (21):

$$\frac{\partial E^{\mu}(\mathbf{w})}{\partial \varsigma_i^{\mu,c+1}} = -(y_{\mu,i} - \varsigma_i^{\mu,c+1}) \tag{27}$$

Incidentally, collecting all the results so far and assuming $c=0$ (no hidden layers), the *delta rule* for non-linear single-layer networks is obtained:

$$\Delta^{\mu} w_{ij} = \alpha \sum_{\mu=1}^{p} (y_{\mu,i} - \varsigma_i^{\mu})\, g'(\hat{\varsigma}_i^{\mu})\, \varsigma_j^{\mu,0} = \alpha \sum_{\mu=1}^{p} \delta^{\mu}_{\ i}\, x_{\mu,j} \tag{28}$$

where the superscript l is dropped (since $c=0$), and $\delta^{\mu}_{\ i} = (y_{\mu,i} - \varsigma_i^{\mu})\, g'(\hat{\varsigma}_i^{\mu})$. Another name for the back-propagation algorithm is *generalized delta rule*. Consequently, for $l \geq 1$, the *deltas* (errors local to a unit) are defined as:

$$\delta^{\mu,l}_{\ i} = \frac{\partial E^{\mu}(\mathbf{w})}{\partial \hat{\varsigma}_i^{\mu,l}} \tag{29}$$

For the output layer $l=c+1$, $\delta^{\mu,l}_{\ i} = (y_{\mu,i} - \varsigma_i^{\mu})\, g'(\hat{\varsigma}_i^{\mu})$. In case g is the logistic, we finally have:

$$\delta^{\mu,l}_{\ i} = (y_{\mu,i} - \varsigma_i^{\mu,c+1})\, \beta\, \varsigma_i^{\mu,l}\,(1-\varsigma_i^{\mu,l}) \tag{30}$$

For the general case $l<c+1$, we proceed as follows:

$$\frac{\partial E^{\mu}(\mathbf{w})}{\partial \varsigma_i^{\mu,l}} = \sum_k \frac{\partial E^{\mu}(\mathbf{w})}{\partial \varsigma_k^{\mu+1,l}} \frac{\partial \varsigma_k^{\mu+1,l}}{\partial \varsigma_i^{\mu,l}} \tag{31}$$

Now $l<c+1$ means there is at least one more layer $l+1$ to the *right* of layer l which is *posterior* in the *feed-forward* computation, but that *precedes l* in the opposite direction. Therefore $\delta E^{\mu}(\underline{w})$ functionally depends on layer $l+1$ *before* than layer l, and the derivative can be split in two. The summation over k is due to the fact that $\delta E^{\mu}(\underline{w})$ depends on every neuron in layer $l+1$. Rewriting (31):

$$\frac{\partial E^{\mu}(\mathbf{w})}{\partial \varsigma_i^{\mu,l}} = \sum_k \frac{\partial E^{\mu}(\mathbf{w})}{\partial \hat{\varsigma}_k^{\mu+1,l}} \frac{\partial \hat{\varsigma}_k^{\mu,l+1}}{\partial \varsigma_i^{\mu,l}} = \sum_k \delta_{\kappa}^{\mu,l+1}\, w_{ki}^{l+1} \tag{32}$$

since

$$\varsigma_k^{\mu,l+1} = \sum_j w_{kj}^{l+1}\, \varsigma_j^{\mu,l}$$

and thus

$$\frac{\partial \hat{\varsigma}_k^{\mu,l+1}}{\partial \varsigma_i^{\mu,l}} = w_{ki}^{l+1}.$$

Putting together (24), (25), (27) and (32) in (23):

$$\Delta^{\mu} w_{ij}^{l} = \frac{\partial E^{\mu}(w)}{\partial \varsigma_i^{\mu,l}} \frac{\partial \varsigma_i^{\mu,l}}{\partial \hat{\varsigma}_i^{\mu,l}} \frac{\partial \hat{\varsigma}_i^{\mu,l}}{\partial w_{ij}^{l}} = \delta^{\mu,l}_{\ i}\, \varsigma_j^{\mu,l-1} \tag{33}$$

where

$$\delta^{\mu,l}_{\ i} = g'(\hat{\varsigma}_i^{\mu,l}) \sum_k \delta^{\mu,l+1}_{\ k}\, w_{ki}^{l+1},$$

the deltas for the hidden units. We show the method (Fig. 1) in algorithmic form, for an *epoch* or presentation of the training set.

Figure 1. Back-propagation algorithm pseudo-code

$for\ all\ \mu\ in\ 1 \leqq \mu \leqq p$

1. *Forward pass. Present* x^{μ} *and compute the outputs* $\varsigma_i^{\mu,l}$ *of all the units.*

2. *Backward pass. Compute the deltas* $\delta^{\mu,l}_i$ *of all the units (the local gradients), as follows:*

$\underline{l=c+1}$: $\delta^{\mu,l}_i = g'(\hat{\varsigma}^{\mu,l}_i)\ (y_{\mu,i} - \varsigma^{\mu,l}_i)$

$\underline{l<c+1}$: $\delta^{\mu,l}_i = g'(\hat{\varsigma}^{\mu,l}_i) \sum_k \delta^{\mu,l+1}_k\ w_{ki}^{l+1}$

3. $\Delta^{\mu} w_{ij}^{l} = \delta^{\mu,l}_i \varsigma^{\mu,l-1}_j$

End

Update weights as $\Delta w_{ij}^{l} = \alpha \sum_{\mu=1}^{p} \Delta^{\mu} w_{ij}^{l}$

CONCLUSION

The strong points of ANNs are the capacity to learn from examples, distributed computation, tolerance to partial failures and the possibility to use them as black-box models. The procedure used to carry out the training process in a neural network is called the training or learning algorithm. Since the objective function is a non linear function of the free parameters, it is not possible to find closed training algorithms for the minima. Preferred algorithms for the multilayer perceptron are the quasi-Newton and Levenberg-Marquardt methods, together with back-propagation to compute the gradient vector efficiently.

REFERENCES

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.

Fletcher, R. (1980). *Practical methods of optimization*. Wiley.

Goldberg, D.E. (1989). *Genetic Algorithms for Search, Optimization & Machine Learning*. Addison-Wesley.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. MacMillan.

Hecht-Nielsen, R. (1990). *Neurocomputing*. Addison-Wesley.

Luenberger, D.J. (1984). *Linear and Nonlinear Programming*. Addison Wesley.

Møller, M. (1993). A scaled conjugate gradient algorithm for supervised learning. *Neural Networks*, 6(4): 525-533.

Parker, D. (1982). Learning logic. Report S81-64, File 1. Office of Technology, Stanford University.

Press, S.A., Teukolsky, Vetterling, W.T., Flannery B.P. (1992). *Numerical recipes in C: The art of scientific computing*. Cambridge University Press.

Rosenblatt, F. (1962). *Principles of neurodynamics*. Spartan Books, NY.

Rumelhart, D., Hinton, G., Williams, R. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1: Foundations). MIT Press, Cambridge.

L

Shewchuck, J.R. (1994). An Introduction to the Conjugate Gradient Method without the Agonizing Pain. School of Computer Science, Carnegie Mellon University.

Toolenaere, T. (1990). Fast adaptive back-propagation with good scaling properties. *Neural Networks*, 3: 561-574, 1990.

Werbos, P.J. (1974). *Beyond regression: new tools for prediction and analysis in the behavioural sciences*. Ph.D. Thesis, Harvard University.

Widrow, B., Hoff, W.H. (1960). Adaptive switching circuits. In Proceedings of the Western Electronic Show and Convention, Vol. 4: 96-104.

KEY TERMS

Artificial Neural Network: Information processing structure without global or shared memory that takes the form of a directed graph where each of the computing elements ("neurons") is a simple processor with internal and adjustable parameters, that operates only when all its incoming information is available.

Back-Propagation: Algorithm for feed-forward multilayer networks that can be used to efficiently compute the gradient vector in all the first-order methods.

Feed-Forward Artificial Neural Network: Artificial Neural Network whose graph has no cycles.

First-Order Method: A training algorithm using the objective function and its gradient vector.

Learning Algorithm: Method or algorithm by virtue of which an Artificial Neural Network develops a representation of the information present in the learning examples, by modification of the weights.

Second-Order Method: A training algorithm using the objective function, its gradient vector and Hessian matrix.

Weight: A free parameter of an Artificial Neural Network, modified through the action of a Learning Algorithm to obtain desired responses to certain input stimuli.

Learning in Feed–Forward Artificial Neural Networks II

Lluís A. Belanche Muñoz
Universitat Politècnica de Catalunya, Spain

INTRODUCTION

Supervised Artificial Neural Networks (ANN) are information processing systems that adapt their functionality as a result of exposure to input-output examples. To this end, there exist generic procedures and techniques, known as *learning rules*. The most widely used in the neural network context rely in derivative information, and are typically associated with the Multilayer Perceptron (MLP). Other kinds of supervised ANN have developed their own techniques. Such is the case of Radial Basis Function (RBF) networks (Poggio & Girosi, 1989). There has been also considerable work on the development of *adhoc* learning methods based on evolutionary algorithms.

BACKGROUND

The problem of learning an input/output relation from a set of examples can be regarded as the task of approximating an unknown function from a set of data points, which are possibly sparse. Concerning approximation by classical feed-forward ANN, these networks implement a parametric approximating function and have been shown to be able of representing generic classes of functions (as the continuous or integrable functions) to an arbitrary degree of accuracy. In general, there are three questions that arise when defining one such parameterized family of functions:

1. What is the most adequate parametric form for a given problem?
2. How to find the best parameters for the chosen form?
3. What classes of functions can be represented and how well?

The most typical problems in a ANN supervised learning process, besides the determination of the learning parameters themselves, include (Hertz, Krogh & Palmer, 1991), (Hinton, 1989), (Bishop, 1995):

1. The possibility of getting stuck in *local optima* of the cost function, in which conventional non-linear optimization techniques will stay forever. The incorporation of a global scheme (like multiple restarts or an annealing schedule) is surely to increase the chance of finding a better solution, although the cost can become prohibitedly high. A feed-forward network has multiple equivalent solutions, created by weight permutations and sign flips. Every local minima in a network with a single hidden layer of h_1 units has $s(h_1)=h_1!2^{h_1}$ equivalent solutions, so the chances of getting in the basin of attraction of one of them are reasonable high. The complexity of the error surface –especially in very high dimensions– makes the possibility of getting trapped a real one.

2. Long *training times*, *oscillations* and network *paralysis*. These are features highly related to the specific learning algorithm, and relate to bad or too general choices for the parameters of the optimization technique (such as the learning rate). The presence of saddle points—regions where the error surface is very flat—also provoke an extremely slow advance for extensive periods of time. The use of more advanced methods that dynamically set these and other parameters can alleviate the problem.

3. *Non-cumulative* learning. It is hard to take an already trained network and re-train it with additional data without losing previously learned knowledge.

4. The *curse of dimensionality*, roughly stated as the fact that the number of examples needed to represent a given function grows exponentially with the number of dimensions.

5. Difficulty of finding a *structure* in the training data, possibly caused by a very high dimension or a distorting pre-processing scheme.

6. Bad *generalization*, which can be due to several causes: the use of poor training data or attempts to extrapolate beyond them, an excessive number of hidden units, too long training processes or a badly chosen regularization. All of them can lead to an *overfitting* of the training data, in which the ANN adjusts the training set merely as an *interpolation* task.

7. Not amenable to *inspection*. It is generally arduous to interpret the knowledge learned, especially in large networks or with a high number of model inputs.

LEARNING IN RBF NETWORKS

A Radial Basis Function network is a type of ANN that can be viewed as the solution of a high-dimensional curve-fitting problem. Learning is equivalent to finding a surface providing the best fit to the data. The RBF network is a two-layered feed forward network using a linear transfer function for the output units and a radially symmetric transfer function for the hidden units. The computation of a hidden unit is expressed as the composition of two functions, as:

$$F_i(x) = \{g(h(x, w_i)), \ w_i \in R^n\}, \quad x \in R^n \qquad (1)$$

with the choice $h(x, w_i) = ||x - w_i||/\theta$ (or other distance measure), with $\theta > 0$ a smoothing term, plus an activation g which very often is a monotonically decreasing response from the origin. These units are localized, in the sense that they give a significant response only in a neighbourhood of their centre w_i. For the activation function a Gaussian $g(z) = exp(-z^2/2)$ is a preferred choice.

Learning in RBF networks is characterized by the separation of the process in two consecutive stages (Haykin, 1994), (Bishop, 1995):

1. Optimize the free parameters of the hidden layer (including the smoothing term) using only the $\{x\}_i$ in D. This is an *unsupervised* method that depends on the *input sample distribution*.

2. With these parameters found and frozen, optimize the $\{c_i\}_i$, the hidden-to-output weights, using the full information in D. This is a *supervised* method that depends on the given *task*.

There are many ways of optimizing the hidden-layer parameters. When the number of hidden neurons equals the number of patterns, each pattern may be taken to be a center of a particular neuron. However, the aim is to form a representation of the probability density function of the data, by placing the centres in only those regions of the input space where significant data are present. One commonly used method is the *k-means algorithm* (McQueen, 1967), which in turn is an approximate version of the maximum-likelihood (ML) solution for determining the location of the means of a mixture density of component densities (that is, maximizing the likelihood of the parameters with respect to the data). The Expectation-Maximization (EM) algorithm (Duda & Hart, 1973) can be used to find the exact ML solution for the means and covariances of the density. It seems that EM is superior to k-means (Nowlan, 1990). The set of centres can also be selected randomly from the set of data points.

The value of the smoothing term can be obtained from the clustering method itself, or else estimated a posteriori. One popular heuristic is:

$$\theta = \frac{d}{\sqrt{2M}} \qquad (2)$$

where d is the maximum distance between the chosen centers and M is the number of centers (hidden units). Alternatively, the method of *Distance Averaging* (Moody and Darken, 1989) can be used, which is the global average over all Euclidean distances between the center of each unit and that of its nearest neighbor.

Once these parameters are chosen and kept constant, assuming the output units are linear, the (square) error function is quadratic, and thus the hidden-to-output weights can be fast and reliably found iteratively by simple gradient descent over the quadratic surface of the error function or directly by solving the minimum norm solution to the over determined least-squares data fitting problem (Orr, 1995).

The whole set of parameters of a RBF network can also be optimized with a global gradient descent procedure on all the free parameters at once (Bishop, 1995), (Haykin, 1994). This brings back the problems of local minima, slow training, etc, already discussed. However, better solutions can in principle be found, because the unsupervised solution focuses on esti-

mating the input probability density function, but the resulting disposition may not be the one minimizing the square error.

EVOLUTIONARY LEARNING ALGORITHMS

The alternative to derivative-based learning algorithms (DBLA) are Evolutionary Algorithms (EA) (Back, 1996). Although the number of successful specific applications of EA is counted by hundreds (see (Back, Fogel & Michalewicz, 1997) for a review), only Genetic Algorithms or GA (Goldberg, 1989) and, to a lesser extent, Evolutionary Programming (Fogel, 1992), have been broadly used for ANN optimization, since the earlier works using genetic algorithms (Montana & Davis, 1989). Evolutionary algorithms operate on a population of individuals applying the principle of *survival of the fittest* to produce better approximations to a solution. At each generation, a new population is created by selecting individuals according to their level of fitness in the problem domain and recombining them using operators borrowed from natural genetics. The offspring also undergo mutation. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

There are comprehensive review papers and guides to the extensive literature on this subject: see (Shaffer, Whitley & Eshelman, 1992), (Yao, 1993), (Kusçu & Thornton, 1994) and (Balakrishnan & Honavar, 1995). One of their main advantages over methods based on derivatives is the global search mechanism. A global method does not imply that the solution is not a local optimum; rather, it eliminates the possibility of getting *caught* in local optima. Another appealing issue is the possibility of performing the traditionally separated steps of determining the best architecture and its weights *at the same time*, in a search over the joint space of structures and weights. Another advantage is the use of potentially any cost measure to assess the goodness of fit or include structural information. Still another possibility is to embody a DBLA into a GA, using the latter to search among the space of structures and the DBLA to optimize the weights; this hybridization leads to extremely high computational costs. Finally, there is the use of EA solely for the numerical optimization problem. In the neural context, this is arguably the task

for which continuous EA are most naturally suited. However, it is difficult to find applications in which GA (or other EA, for that matter) have clearly outperformed DBLA for supervised training of feed-forward neural networks (Whitley, 1995). It has been pointed out that this task is inherently hard for algorithms that rely heavily on the recombination of potential solutions (Radcliffe, 1991). In addition, the training times can become too costly, even worse than that for DBLA.

In general, Evolutionary Algorithms –particularly, the continuous ones– are in need of specific research devoted to ascertain their general validity as alternatives to DBLA in neural network optimization. Theoretical as well as practical work, oriented to tailor specific EA parameters for this task, together with a specialized operator design should pave the way to a fruitful assessment of validity.

FUTURE TRENDS

Research in ANN currently concerns the development of *learning algorithms* for weight adaptation or, more often, the enhancement of existing ones. New *architectures* (ways of arranging the units in the network) are also introduced from time to time. Classical *neuron models*, although useful and effective, are lessened to a few generic function classes, of which only a handful of instances are used in practice.

One of the most attractive enhancements is the extension of neuron models to modern data mining situations, such as data heterogeneity. Although a feed-forward neural network can in principle approximate an arbitrary function to any desired degree of accuracy, in practice a pre-processing scheme is often applied to the data samples to ease the task. In many important domains from the real world, objects are described by a mixture of continuous and discrete variables, usually containing missing information and characterized by an underlying vagueness, uncertainty or imprecision. For example, in the well-known UCI repository (Murphy & Aha, 1991) over half of the problems contain *explicitly declared* nominal attributes, let alone other discrete types or fuzzy information, usually unreported. This heterogeneous information should not be treated in general as real-valued quantities. Conventional ways of encoding non-standard information in ANN include (Prechelt, 1994), (Bishop, 1995), (Fiesler & Beale, 1997):

Ordinal variables. These variables correspond to discrete (finite) sets of values wherein an ordering has been defined (possibly only partial). They are more than often treated as real-valued, and mapped equidistantly on an arbitrary real interval. A second possibility is to encode them using a *thermometer*. To this end, let k be the number of ordered values; k new *binary* inputs are then created. To represent value i, for $1 \leq i \leq k$, the leftmost $1,...,i$ units will be on, and the remaining $i+1,...,k$ off.

The interest in these variables relies in that they appear frequently in real domains, either as symbolic information or from processes that are discrete in nature. Note that an ordinal variable need not be numerical.

Nominal variables Nominal variables are unanimously encoded using a 1-out-of-k representation, being k the number of values, which are then encoded as the rows of the $I_{k \times k}$ identity matrix.

Missing values Missing information is an old issue in statistical analysis (Little & Rubin, 1987). There are several causes for the absence of a value. They are very common in Medicine and Engineering, where many variables come from on-line sensors or device measurements. Missing information is difficult to handle, especially when the lost parts are of significant size. It can be either removed (the entire case) or "filled in" with the mean, median, nearest neighbour, or encoded by adding another input equal to one only if the value is absent and zero otherwise. Statistical approaches need to make assumptions about or model the input distribution itself. The main problem with missing data is that we never know if all the efforts devoted to their estimation will revert, in practice, in better-behaved data. This is also the reason why we develop on the treatment of missing values as part of the general discussion on data characteristics. The reviewed methods pre-process the data to make it acceptable by models that otherwise would not accept it. In the case of missing values, the data is *completed* because the available neural methods only admit complete data sets.

Uncertainty. Vagueness, imprecision and other sources of uncertainty are considerations usually put aside in the ANN paradigm. Nonetheless, many variables in learning processes are likely to bear some form of uncertainty. In Engineering, for example, on-line sensors are likely to get old with time and continuous use, and this may be reflected in the quality of their measurements. In many occasions, the data at hand are imprecise for a manifold of reasons: technical limitations, a veritable qualitative origin, or even we can be interested in introducing imprecision with the purpose of augmenting the capacity for abstraction or generalization (Esteva, Godo & García), possibly because the underlying process is believed to be less precise than the available measures.

In Fuzzy Systems theory there are explicit formalisms for representing and manipulating uncertainty, that is precisely what the system best models and manages. It is perplexing that, when supplying this kind of input/output data, we require the network to approximate the desired output in a very precise way. Sometimes the known value takes an interval form: "between 5.1 and 5.5", so that any transformation to a real value will result in a loss of information. A more common situation is the absence of numerical knowledge. For example, consider the value "fairly tall" for the variable *height*. Again, Fuzzy Systems are comfortable, but for an ANN this is real trouble. The integration of symbolic and continuous information is also important because numeric methods bring higher *concretion*, whereas symbolic methods bring higher *abstraction*. Their combined use is likely to increase the flexibility of hybrid systems. For numeric data, an added flexibility is obtained by considering imprecision in their values, leading to fuzzy numbers (Zimmermann, 1992).

CONCLUSION

As explained at length in other chapters, derivative-based learning algorithms make a number of assumptions about the local error surface and its differentiability. In addition, the existence of local minima is often neglected or overlooked entirely. In fact, the possibility of getting caught in these minima is more than often circumvented by multiple runs of the algorithm (that is, multiple restarts from different initial points in weight space). This "sampling" procedure is actually an implementation of a very naïve stochastic process. A global training algorithm for neural networks is the evolutionary algorithm, a stochastic search training algorithm based on the mechanics of natural genetics and biological evolution. It requires information from the objective function, but not from the gradient vector or the Hessian matrix and thus it is a zero-order method. On the other hand, there is an emerging need to devise neuron models that properly handle different data types,

as is done in support vector machines (Shawe-Taylor & Cristianini, 2004), where kernel design is a current research topic.

REFERENCES

Balakrishnan, K., Honavar, V. (1995). Evolutionary design of neural architectures - a preliminary taxonomy and guide to literature. Technical report CS-TR-95-01. Dept. of Computer Science. Iowa State University.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford Univ. Press, New York

Bäck, Th., Fogel D.B., Michalewicz, Z. (Eds., 1997) *Handbook of Evolutionary Computation*. IOP Publishing & Oxford Univ. Press.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.

Esteva, F., Godo, L., García, P. (1998). Similarity-based Reasoning. IIIA Research Report 98-31, Instituto de Investigación en Inteligencia Artificial. Barcelona, Spain.

Duda, R.O., Hart, P.E. (1973) *Pattern classification and scene analysis*. John Wiley.

Fiesler, E., Beale, R. (Eds., 1997) *Handbook of Neural Computation*. IOP Publishing & Oxford Univ. Press.

Fogel, L.J. (1992). An analysis of evolutionary programming. In Fogel and Atmar (Eds.) Procs. of the 1st annual conf. on evolutionary programming. La Jolla, CA: Evolutionary Programming Society.

Goldberg, D.E. (1989). *Genetic Algorithms for Search, Optimization & Machine Learning*. Addison-Wesley.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. MacMillan.

Hertz, J., Krogh, A., Palmer R.G. (1991). *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City.

Hinton, G.E. (1989). Connectionist learning procedures. *Artificial Intelligence*, 40: 185-234.

Kusçu, I., Thornton, C. (1994). Design of Artificial Neural Networks using genetic algorithms: review and prospect. Technical Report of the Cognitive and Computing Sciences Dept. University of Sussex, England.

Little, R.J.A., Rubin, D.B. (1987). *Statistical analysis with missing data*. John Wiley.

McQueen, J. (1967) Some methods of classification and analysis of multivariate observations. In Procs. of the 5th Berkeley Symposium on Mathematics, Statistics and Probability. LeCam and Neyman (eds.), University of California Press.

Montana, D.J., Davis, L. (1989). Training Feed-Forward Neural Networks using Genetic Algorithms. In Proceedings of the 11th International Joint Conference on Artificial Intelligence. Morgan Kaufmann.

Moody, J. and Darken, C. (1989): J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. Neural Computation, 1:281-294.

Murphy, P.M., Aha, D. (1991). UCI Repository of machine learning databases. UCI Dept. of Information and Computer Science.

Nowlan, S. (1990). Max-likelihood competition in RBF networks. Technical Report CRG-TR-90-2. Connectionist Research Group, Univ. of Toronto.

Orr, M.J.L. (1995) Introduction to Radial Basis Function Networks. Technical Report of the Centre for Cognitive Science, Univ. of Edinburgh.

Poggio T., Girosi, F. (1989). A Theory of Networks for Approximation and Learning. AI Memo No. 1140, AI Laboratory, MIT.

Prechelt, L. (1994). Proben1: A set of Neural Network Benchmark Problems and Benchmarking Rules. Technical Report 21/94. Universität Karlsruhe.

Radcliffe, N.J. (1991). Genetic set recombination and its application to neural network topology optimization. Technical Report EPCC-TR-91-21. University of Edinburgh.

Shaffer, J.D., Whitley, D., Eshelman, (1992). L.J. Combination of Genetic Algorithms and Neural Networks: A Survey of the State of the Art. In *Combination of Genetic Algorithms and Neural Networks*. Shaffer, J.D., Whitley, D. (eds.), pp. 1-37.

Shawe-Taylor, J. Cristianini, N. (2004). Kernel Methods for Pattern Analysis. Cambridge University Press.

Whitley, D. (1995). Genetic Algorithms and Neural Networks. In *Genetic Algorithms in Engineering and Computer Science*. Periaux, Galán, Cuesta (eds.), John Wiley.

Yao, X. (1993). A Review of Evolutionary Artificial Networks. *Intl. Journal of Intelligent Systems*, 8(4): 539-567, 1993.

Zimmermann, H.J. (1992). *Fuzzy set theory and its applications*. Kluver.

KEY TERMS

Architecture: The number of artificial neurons, its arrangement and connectivity.

Artificial Neural Network: Information processing structure without global or shared memory that takes the form of a directed graph where each of the computing elements ("neurons") is a simple processor with internal and adjustable parameters, that operates only when all its incoming information is available.

Evolutionary Algorithm: A computer simulation in which a population of *individuals* (abstract representations of candidate solutions to an optimization problem) are stochastically selected, recombined, mutated, and then removed or kept, based on their relative fitness to the problem.

Feed-Forward Artificial Neural Network: Artificial Neural Network whose graph has no cycles.

Learning Algorithm: Method or algorithm by virtue of which an Artificial Neural Network develops a representation of the information present in the learning examples, by modification of the weights.

Neuron Model: The computation of an artificial neuron, expressed as a function of its input and its weight vector and other local information.

Weight: A free parameter of an Artificial Neural Network, that can be modified through the action of a Learning Algorithm to obtain desired responses to certain input stimuli.

L

Learning Nash Equilibria in Non–Cooperative Games

Alfredo Garro
University of Calabria, Italy

INTRODUCTION

Game Theory (Von Neumann & Morgenstern, 1944) is a branch of applied mathematics and economics that studies situations (games) where self-interested interacting players act for maximizing their returns; therefore, the return of each player depends on his behaviour and on the behaviours of the other players. Game Theory, which plays an important role in the social and political sciences, has recently drawn attention in new academic fields which go from algorithmic mechanism design to cybernetics. However, a fundamental problem to solve for effectively applying Game Theory in real word applications is the definition of well-founded solution concepts of a game and the design of efficient algorithms for their computation.

A widely accepted solution concept of a game in which any cooperation among the players must be self-enforcing (non-cooperative game) is represented by the Nash Equilibrium. In particular, a Nash Equilibrium is a set of strategies, one for each player of the game, such that no player can benefit by changing his strategy unilaterally, i.e. while the other players keep their strategies unchanged (Nash, 1951). The problem of computing Nash Equilibria in non-cooperative games is considered one of the most important open problem in Complexity Theory (Papadimitriou, 2001). Daskalakis, Goldbergy, and Papadimitriou (2005), showed that the problem of computing a Nash equilibrium in a game with four or more players is complete for the complexity class PPAD-Polynomial Parity Argument Directed version (Papadimitriou, 1991), moreover, Chen and Deng extended this result for 2-player games (Chen & Deng, 2005). However, even in the two players case, the best algorithm known has an exponential worst-case running time (Savani & von Stengel, 2004); furthermore, if the computation of equilibria with simple additional properties is required, the problem immediately becomes NP-hard (Bonifaci, Di Iorio, & Laura, 2005) (Conitzer & Sandholm, 2003) (Gilboa & Zemel, 1989) (Gottlob, Greco, & Scarcello, 2003).

Motivated by these results, recent studies have dealt with the problem of efficiently computing Nash Equilibria by exploiting approaches based on the concepts of learning and evolution (Fudenberg & Levine, 1998) (Maynard Smith, 1982). In these approaches the Nash Equilibria of a game are not statically computed but are the result of the evolution of a system composed by agents playing the game. In particular, each agent after different rounds will learn to play a strategy that, under the hypothesis of agent's rationality, will be one of the Nash equilibria of the game (Benaim & Hirsch, 1999) (Carmel & Markovitch, 1996).

This article presents SALENE, a Multi-Agent System (MAS) for learning Nash Equilibria in non-cooperative games, which is based on the above mentioned concepts.

BACKGROUND

An *n*-person strategic game G can be defined as a tuple $G = (N; (A^i)_{i \in N}; (r^i)_{i \in N})$, where $N = \{1, 2, \ldots, n\}$ is the set of players, A^i is a finite set of actions for player $i \in N$, and $r^i : A^1 \times \ldots \times A^n \to \Re$ is the payoff function of player i. The set A^i is called also the set of pure strategies of player i. The Cartesian product $\times_{i \, N} A^i = A^1 \times \ldots \times A^n$ can be denoted by A and $r : A \to \Re^N$ can denote the vector valued function whose *i*th component is r^i, i.e., $r(a) = (r^1(a), \ldots, r^n(a))$, so it is possible to write (N, A, r) for short for $(N; (A^i)_{i \in N}; (r^i)_{i \in N})$.

For any finite set A^i the set of all probability distributions on A^i can be denoted by $\Delta(A^i)$. An element $\sigma^i \in \Delta(A^i)$ is a mixed strategy for player i.

A (Nash) equilibrium of a strategic game $G = (N, A, r)$ is an N-tuple of (mixed) strategies $\sigma = (\sigma^i)_{i \in N}$, $\sigma^i \in \Delta(A^i)$, such that for every $i \in N$ and any other strategy of player i, $\tau^i \in \Delta(A^i)$, $r^i(\tau^i, \sigma^{-i}) \leq r^i(\sigma^i, \sigma^{-i})$, where r^i denotes also the expected payoff to player i in the mixed extension of the game and σ^{-i} represents the mixed strategies in σ of all the other players. Basically, supposing that all the other players do not change their

strategies it is not possible for any player *i* to play a different strategy τ^i able to gain a better payoff of that gained by playing σ^i. σ^i is called a Nash equilibrium strategy for player *i*.

In 1951 J. F. Nash proved that a strategic (non-cooperative) game $G = (N, A, r)$ has at least a (Nash) equilibrium σ (Nash, 1951); in his honour, the computational problem of finding such equilibria is known as NASH (Papadimitriou, 1994).

SOFTWARE AGENTS FOR LEARNING NASH EQUILIBRIA

SALENE was conceived as a system for learning at least one Nash Equilibrium of a non-cooperative game given in the form $G = (N; (A^i)_{i \in N}; (r^i)_{i \in N})$. In particular, the system asks the user for:

- the number *n* of the players which defines the set of players $N = \{1, 2, \ldots , n\}$;
- for each player $i \in N$, the related finite set of pure strategies A^i and his payoff function $r^i : A^1 \times \ldots \times A^n \to \Re$;
- the number *k* of times the players will play the game.

Figure 1. The class diagram of SALENE
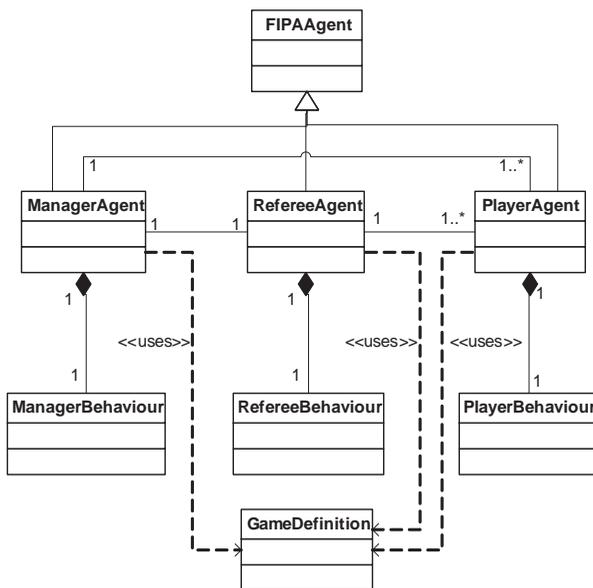
Then, the system creates *n* agents, one associated to each player, and a referee. The agents will play the game *G k* times, after each match, each agent will decide the strategy to play in the next match to maximise his expected utility on the basis of his beliefs about the strategies that the other agents are adopting. By analyzing the behaviour of each agent in all the *k* matches of the game, SALENE presents to the user an estimate of a Nash Equilibrium of the game. The Agent paradigm has represented a "natural" way of modelling and implementing the proposed solution as it is characterized by several interacting autonomous entities (players) which try to achieve their goals (consisting in maximising their returns).

The class diagram of SALENE is shown in Figure 1.

The Manager Agent interacts with the user and it is responsible for the global behaviour of the system. In particular, after having obtained from the user the input parameters *G* and *k*, the Manager Agent creates both *n* Player Agents and a Referee Agent that coordinates and monitors the behaviours of the players. The Manager Agent sends to all the agents the definition *G* of the game then he asks the Referee Agent to orchestrate *k* matches of the game *G*. In each match, the Referee Agent asks each Player Agent which pure strategy he has decided to play, then, after having acquired the strategies from all players, the Referee Agent communicates to each Player Agent both the strategies played and the payoffs gained by all players. After playing *k* matches of the game *G* the Referee Agent communicates all the data about the played matches to the Manager Agent which analyses it and properly presents the obtained results to the user.

A Player Agent is a rational player that, given the game definition *G*, acts to maximise his expected utility in each single match of *G* without considering the overall utility that he could obtain in a set of matches. In particular the behaviour of the Player Agent *i* can be described by the following main steps:

1. In the first match the Player Agent *i* chooses to play a pure strategy randomly generated considering all the pure strategies playable with the same probability: if $|A^i| = m$ the probability of choosing a pure strategy $s \in A^i$ is $1/m$;
2. The Player Agent *i* waits for the Referee Agent to ask him which strategy he wants to play, then he communicates to the Referee Agent the chosen

pure strategy as computed in step 1 if he is playing his first match or in step 4 otherwise;

3. The Player Agent waits for the Referee Agent to communicate him both the pure strategies played and the payoffs gained by all players;

4. The Player Agent decides the mixed strategy to play in the next match. In particular, the Player Agent updates the beliefs about the mixed strategies currently adopted by the other players and consequently recalculate the strategy able to maximise his expected utility. Basically, the Player Agent i tries to find the strategy $\sigma^i \in \Delta(A^i)$, such that for any other strategy $\tau^i \in \Delta(A^i)$, $r^i(\tau^i,\sigma^{-i}) \leq r^i(\sigma^i,\sigma^{-i})$ where r^i denotes his expected payoff and σ^{-i} represents his beliefs about the mixed strategies currently adopted by all the other players, i.e. $\sigma^{-i}=(\sigma^j)_{j\in N, j\neq i}$, $\sigma^j \in \Delta(A^j)$. In order to evaluate σ^j for each other player $j\neq i$ the Player Agent i considers the pure strategies played by the player j in all the previous matches and computes the frequency of each pure strategy, this frequency distribution will be the estimate for σ^j. If there is at least an element in the actually computed set $\sigma^{-i}=(\sigma^j)_{j\in N, j\neq i}$ that differs from the set σ^{-i} as computed in the previous match, the Player Agent i solves the inequality $r^i(\tau^i,\sigma^{-i}) \leq r^i(\sigma^i,\sigma^{-i})$ that is equivalent to solve the optimization problem $P=\{\max(r^i(\sigma^i,\sigma^{-i})), \sigma^i\in\Delta(A^i)\}$. It is worth noting that P is a linear optimization problem, actually, given the set σ^{-i}, $r^i(\sigma^i,\sigma^{-i})$ is a linear objective function in σ^i (see the game definition reported in the Background Section), and with $|A^i|=m$ $\sigma^i\in\Delta(A^i)$ is a vector $\chi \in \Re^M$ such that $\Sigma_{s\ M}\chi_s=1$ and for every $s\in M$ $\chi_s\geq0$, so the constraint $\sigma^i\in\Delta(A^i)$ is a set of $m+1$ linear inequalities. P is solved by the Player Agent by using an efficient method for solving problems in linear programming, in particular the predictor-corrector method of Mehrotra (1992), whose complexity is polynomial for both average and worst case. The obtained solution for σ^i is a pure strategy because it is one of the vertices of the polytope which defines the feasible region for P. The obtained strategy σ^i will be played by the Player Agent i in the next match; $r^i(\sigma^i,\sigma^{-i})$ represents the expected payoff to player i in the next match;

5. back to step 2.

It is worth noting that a Player Agent for choosing the mixed strategy to play in each match of G does not need to known the payoff functions of the others players, in fact, for solving the optimization problem P it only needs to consider the strategies which have been played by the other players in all the previous matches.

The Manager Agent, receives from the Referee Agent all the data about the k matches of the game G and computes an estimate of a Nash Equilibrium of G, i.e. an N-tuple $\sigma=(\sigma^i)_{i\in N}$, $\sigma^i\in\Delta(A^i)$. In particular, in order to estimate σ^i (the Nash equilibrium strategy of the player i), the Manager Agent computes, on the basis of the pure strategies played by the player i in each of the k match, the frequency of each pure strategy: this frequency distribution will be the estimate for σ^i. The so computed set $\sigma=(\sigma^i)_{i\in N}$, $\sigma^i\in\Delta(A^i)$ will be then properly proposed to the user together with the data exploited for its estimation.

SALENE has been implemented using JADE (Bellifemine, Poggi, & Rimassa, 2001), a software framework allowing for the development of multi-agent systems and applications conforming to FIPA standards (FIPA, 2006), and tested on different games that differ from each other both in the number and in the kind of Nash Equilibria. The experiments have demonstrated that:

- if the game has $p>=1$ Pure Nash Equilibria and $s>=0$ Mixed Nash Equilibria the agents converge in playing one of the p Pure Nash Equilibria; in these cases, as the behaviour of each Player Agent converges with probability one to a Nash Equilibrium of the game, the learning process *converges in behaviours to equilibrium* (Foster & Young, 2003);

- if the game has only Mixed Nash Equilibria, while the behaviour of the Player Agents does not converge to an equilibrium, the time-average behaviour, i.e. the empirical frequency with which each player chooses his strategy, may converge to one of the mixed Nash Equilibria of the game; that is the learning process may *converge in time average to equilibrium* (Foster and Young, 2003).

In the next Section the main aspects related to the convergence properties of the approach/algorithm

L

exploited by the SALENE agents for leaning Nash Equilibria are discussed in a more general discussion about current and future research efforts.

FUTURE TRENDS

Innovative approaches, as SALENE, based on the concepts of learning and evolution have shown great potential for modelling and efficiently solving non-cooperative games. However, as the solutions of the games (e.g. Nash Equilibria) are not statically computed but are the result of the evolution of a system composed by interacting agents, there are several open problems mainly related to the accuracy of the provided solution that need to be tackled to allow these approaches to be widely exploited in concrete business application.

The approach exploited in SALENE, which derives from the *Fictitious Play* (Robinson, 1951) approach, efficiently solves the problem of learning a Nash Equilibrium in non-cooperative games which have at least one Pure Nash Equilibrium: in such a case the behaviour of the players exactly converges to one of the Pure Nash Equilibria of the game (*convergence in behaviours to equilibrium*). On the contrary, if the game has only Mixed Nash Equilibria, the convergence of the learning algorithm is not ensured. Computing *ex ante* when this case happens is quite costly as it requires to solve the following problem: "Determining whether a strategic game has only Mixed Nash Equilibria", which is equivalent to: "Determining whether a strategic game does not have any Pure Nash Equilibria". This problem is Co-NP complete as its complement "Determining whether a strategic game has a Pure Nash Equilibrium" is NP complete (Gottlob, Greco, & Scarcello, 2003). As witnessed by the conducted experiments, when a game has only Mixed Nash Equilibria there are still some cases in which, while the behaviour of the players does not converge to an equilibrium, the time-average behaviour, i.e. the empirical frequency with which each player chooses his strategy, converges to one of the Mixed Nash Equilibria of the game (*convergence in time average to equilibrium*).

Nevertheless, there are some cases in which there is neither *convergence in behaviour* neither *convergence in time average to equilibrium*; an example of such a case is the *fashion game* of Shapley (1964). An important open problem is then represented by the characterization of the classes of games for which the learning algorithm

adopted in SALENE converges; more specifically, the classes of games for which the algorithm: (a) *convergences in behaviours to equilibrium* (which implies the *convergence in time average to equilibrium*), (b) only *convergences in time average to equilibrium*; (c) does not converge neither *in behaviours* neither in *time average*. Currently, it has been demonstrated that the algorithm converges *in behaviours* or in *time average to equilibrium* for the following classes of games:

- zero-sum games (Robinson, 1951);
- games which are solvable by iterated elimination of strictly dominated strategies (Nachbar, 1990);
- potential games (Monderer & Shapley, 1996);
- 2xN games, i.e. games with 2 players, 2 strategies for one player and N strategies for the other player (Berger, 2005).

Future efforts will be geared towards: (i) completing the characterization of the classes of games for which the learning algorithm adopted in SALENE converges and evaluating the complexity of solving the *membership problem* for such a classes; (ii) evaluating different learning algorithms and their convergence properties; (ii) letting the user ask for the computation of Nash Equilibria with simple additional properties.

More in general, a wide adoption of the emerging agent-based approaches for solving games which model concrete business applications will depend on the accuracy and the convergence properties of the provided solutions; both aspects still need to be fully investigated.

CONCLUSION

The complexity of NASH, the problem consisting in computing Nash Equilibria in non-cooperative games, is still debated, but even in the two players case, the best known algorithm has an exponential worst-case running time. SALENE, the proposed MAS for learning Nash Equilibria in non-cooperative games, can be conceived as a heuristic and efficient method for computing at least one Nash Equilibria in a non-cooperative game represented in its normal form; actually, the learning algorithm adopted by the Player Agents has a polynomial running time for both average and worst case. SALENE can be then fruitfully exploited for

efficiently solving non-cooperative games which model interesting concrete problems ranging from classical economic and finance problems to the emerging ones related to the economic aspects of the Internet such as TCP/IP congestion, selfish routing, and algorithmic mechanism design.

REFERENCES

Bellifemine, F., Poggi, A., & Rimassa, G. (2001). Developing multi-agent systems with a FIPA-compliant agent framework. *Software Practice and Experience*, 31(2),103-128.

Benaim, M., & Hirsch, M.W. (1999). Learning process, mixed equilibria and dynamic system arising for repeated games. *Games and Economic Behavior*, 29, 36-72.

Berger, U. (2005). Fictitious Play in 2xN Games. *Journal of Economic Theory*, 120, 139-154.

Bonifaci, V., Di Iorio, U., & Laura, L. (2005). On the complexity of uniformly mixed Nash equilibria and related regular subgraph problems. *Proceedings of the 15th International Symposium on Fundamentals of Computation Theory*, 197-208.

Carmel, D., & Markovitch, S. (1996). Learning Models of Intelligent Agents. *Proceedings of the 13th National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, Vol. 2, 62-67, Menlo Park, California: AAAI Press.

Chen, X., & Deng, X. (2005). Settling the Complexity of 2-Player Nash-Equilibrium. *Electronic Colloquium on Computational Complexity*, Report No. 140.

Conitzer, V., & Sandholm, T. (2003). Complexity results about Nash equilibria. *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 765-771.

Daskalakis, C., Goldbergy, P.W., & Papadimitriou, C.H. (2005). The Complexity of Computing a Nash Equilibrium. *Electronic Colloquium on Computational Complexity*, Report No. 115.

FIPA (2006). Foundation for Intelligent Physical Agents, *http://www.fipa.org*.

Foster, D.P., & Young, P.H. (2003). Learning, Hypothesis Testing, and Nash Equilibrium. *Games and Economic Behavior*, 45(1), 73-96.

Fudenberg, D., & Levine, D. (1998). The Theory of Learning in Games. Cambridge, MA: MIT Press.

Gilboa, I., & Zemel, E. (1989). Nash and correlated equilibria: some complexity considerations. *Games and Economic Behavior*, 1(1), 80-93.

Gottlob, G., Greco, G., & Scarcello, F. (2003). Pure Nash Equilibria: hard and easy games. *Proceedings of the 9th Conference on Theorethical Aspects of Rationality and Knowledge*, 215-230.

Maynard Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge University Press.

Mehrotra, S. (1992). On the Implementation of a Primal-dual Interior Point Method. *SIAM Optimization Journal*, 2, 575-601.

Monderer, D., & Shapley, L.S. (1996). Fictitious Play Property for Games with Identical Interests. *Journal of Economic Theory*, 68, 258-265.

Nachbar, J. (1990). Evolutionary Selection Dynamics in Games: Convergence and Limit Properties. *International Journal of Game Theory*, 19, 59-89.

Nash, J.F. (1951). Non-cooperative games. *Annals of Mathematics*, 54, 289-295.

Papadimitriou, C.H. (1991). On inefficient proofs of existence and complexity classes. *Proceedings of the 4th Czechoslovakian Symposium on Combinatorics*.

Papadimitriou, C.H. (1994). On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and Systems Sciences*, 48(3), 498-532.

Papadimitriou, C.H. (2001). Algorithms, Games and the Internet. *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*, 749-753.

Robinson, J. (1951). An Iterative Method of Solving a Game. *Annals of Mathematics*, 54, 296-301.

Savani, R., & von Stengel, B. (2004). Exponentially many steps for finding a Nash equilibrium in a bimatrix game. *Proceedings of the 45th Symposium on Foundations of Computer Science*, 258-267.

L

Shapley, L.S. (1964). Some topics in two-person games. *Advances in Game Theory*, 1-28, Dresher, M., Shapley, L.S., & Tucker, A. W. editors, Princeton University Press.

Tsebelis, G. (1990). *Nested Games: rational choice in comparative politics*. University of California Press.

Von Neumann, J, & Morgenstern, O. (1944). *Theory of Games and economic Behaviour*. Princeton University Press.

KEY TERMS

Computational Complexity Theory: A branch of the theory of computation in computer science which studies how the running time and the memory requirements of an algorithm increase as the size of the input to the algorithm increases.

Game Theory: A branch of applied mathematics and economics that studies situations (games) where self-interested interacting players act for maximizing their returns.

Heuristic: In computer science, a technique designed to solve a problem which allows for gaining computational performance or conceptual simplicity potentially at the cost of accuracy and/or precision of the provided solutions to the problem itself.

Nash Equilibrium: A solution concept of a game where no player can benefit by changing his strategy unilaterally, i.e. while the other players keep theirs unchanged; this set of strategies and the corresponding payoffs constitute a Nash Equilibrium of the game.

NP-Hard Problems: Problems that are intrinsically harder than those that can be solved by a nondeterministic Turing machine in polynomial time.

Non-Cooperative Games: A game in which any co-operation among the players must be self-enforcing.

Payoffs: Numeric representations of the utility obtainable by a player in the different outcomes of a game.

Learning-Based Planning

Sergio Jiménez Celorrio
Universidad Carlos III de Madrid, Spain

Tomás de la Rosa Turbides
Universidad Carlos III de Madrid, Spain

INTRODUCTION

Automated Planning (AP) studies the generation of action sequences for problem solving. A problem in AP is defined by a state-transition function describing the dynamics of the world, the initial state of the world and the goals to be achieved. According to this definition, AP problems seem to be easily tackled by searching for a path in a graph, which is a well-studied problem. However, the graphs resulting from AP problems are so large that explicitly specifying them is not feasible. Thus, different approaches have been tried to address AP problems. Since the mid 90's, new planning algorithms have enabled the solution of practical-size AP problems. Nevertheless, domain-independent planners still fail in solving complex AP problems, as solving planning tasks is a PSPACE-Complete problem (Bylander, 94).

How do humans cope with this planning-inherent complexity? One answer is that our experience allows us to solve problems more quickly; we are endowed with learning skills that help us plan when problems are selected from a stable population. Inspire by this idea, the field of learning-based planning studies the development of AP systems able to modify their performance according to previous experiences.

Since the first days, Artificial Intelligence (AI) has been concerned with the problem of Machine Learning (ML). As early as 1959, Arthur L. Samuel developed a prominent program that learned to improve its play in the game of checkers (Samuel, 1959). It is hardly surprising that ML has often been used to make changes in systems that perform tasks associated with AI, such as perception, robot control or AP. This article analyses the diverse ways ML can be used to improve AP processes. First, we review the major AP concepts and summarize the main research done in learning-based planning. Second, we describe current trends in applying ML to AP. Finally, we comment on the next avenues for combining AP and ML and conclude.

BACKGROUND

The languages for representing AP tasks are typically based on extensions of first-order logic. They encode tasks using a set of actions that represents the state-transition function of the world (the planning domain) and a set of first-order predicates that represent the initial state together with the goals of the AP task (the planning problem). In the early days of AP, STRIPS was the most popular representation language. In 1998 the Planning Domain Definition Language (PDDL) was developed for the first International Planning Competition (IPC) and since that date it has become the standard language for the AP community. In PDDL (Fox & Long, 2003), an action in the planning domain is represented by: (1) the action preconditions, a list of predicates indicating the facts that must be true so the action becomes applicable and (2) the action post-conditions, typically separated in *add* and *delete* lists, which are lists of predicates indicating the changes in the state after the action is applied.

Before the mid '90s, automated planners could only synthesize plans of no more than 10 actions in an acceptable amount of time. During those years, planners strongly depended on speedup techniques for solving AP problems. Therefore, the application of search control became a very popular solution to accelerate planning algorithms. In the late 90's, a significant scale-up in planning took place due to the appearance of the reachability planning graphs (Blum & Furst, 1995) and the development of powerful domain independent heuristics (Hoffman & Nebel, 2001) (Bonet & Geffner, 2001). Planners using these approaches could often synthesize 100-action plans just in seconds.

At the present time, there is not such dependence on ML for solving AP problems, but there is a renewed interest in applying ML to AP motivated by three factors: (1) IPC-2000 showed that knowledge-based planners significantly outperform domain-independent planners. The development of ML techniques that automatically define the kind of knowledge that humans put in these planners would bring great advances to the field. (2) Domain-independent planners are still not able to cope with real-world complex problems. On the contrary, these problems are often solved by defining ad hoc planning strategies by hand. ML promises to be a solution to automatically defining these strategies. And, (3) there is a need for tools that assist in the definition, validation and maintenance of planning-domain models. At the moment, these processes are still done by hand.

LEARNING-BASED PLANNING

This section describes the current ML techniques for improving the performance of planning systems. These techniques are grouped according to the target of learning: search control, domains-specific planners, or domain models.

Learning Search Control

Domain-independent planners require high search effort, so search-control knowledge is frequently used to reduce this effort. Hand-coded control knowledge has proved to be useful in many domains, however is difficult for humans to formalize it, as it requires specific knowledge of the planning domains and the planner structure. Since AP's early days, diverse ML techniques have been developed with the aim of automatically learning search-control knowledge. A few examples of these techniques are macro-actions (Fikes, Hart & Nilsson, 1972), control-rules (Borrajo & Veloso, 1997), and case-based and analogical planning (Veloso, 1994).

At the present, most of the state-of-the-art planners are based on heuristic search over the state space (12 of the 20 participants in IPC-2006 used this approach). These planners achieve impressive performance in many domains and problems, but their performance strongly depends on the definition of a good domain-independent heuristic function. These heuristics are computed solving a simplified version of the planning

task, which ignores the delete list of actions. The solution to the simplified task is taken as the estimated cost for reaching the task goals. These kinds of heuristics provide good guidance across the wide range of different domains. However, they have some faults: (1) in many domains, these heuristic functions vastly underestimate the distance to the goal leading to poor guidance, (2) the computation of the heuristic values of the search nodes is too expensive, and (3) these heuristics are non-admissible so heuristics planners do not find good solutions in terms of plan quality.

Since evaluating a search node in heuristic planning is so time consuming, (De la Rosa, García-Olaya & Borrajo, 2007) proposed using Case-based Reasoning (CBR) to reduce the number of explored nodes. Their approach stores sequences of abstracted state transitions related to each particular object in a problem instance. Then, with a new problem, these sequences are retrieved and re-instantiated to support a forward heuristic search, deciding the node ordering for computing its heuristic value.

In the last years, other approaches have been developed to minimize the negative effects of the heuristic through ML: (Botea, Enzenberger, Müller & Schaeffer, 2005) learned off-line macro-actions to reduce the number of evaluated nodes by decreasing the depth of the search tree. (Coles & Smith, 2007) learned on-line macro-actions to escape from plateaus in the search tree without any exploration. (Yoon, Fern & Givan, 2006) proposed using an inductive approach to correct the domain-independent heuristic in those domains based on learning a supplement to the heuristic from observations of solved problems in these domains.

All these methods for learning search-control knowledge suffer from the utility problem. Learning too much control knowledge can actually be counterproductive because the difficulty of storing and managing the information and the difficulty of determining which information to use when solving a particular problem can interfere with efficiency.

Learning Domain-Specific Planners

An alternative approach to learning search control consists of learning domain-specific planning programs. These programs receive as input a planning problem of a fixed domain and return a plan that solves the problem.

The first approaches to learn domain-specific planners were based on supervised inductive learning; they used genetic programming (Spector, 1994) and decision-list learning (Khardon, 1999), but they were not able to reliably produce good results. Recently, (Winner & Veloso, 2003) presented a different approach based on generalizing an example plan into a domain-specific planning program and merging the resulting source code with the previous ones.

Domain-specific planners are also represented as policies, i.e., pairs of state and the preferred action to be executed in the state. Relational Reinforcement Learning (RRL) (Dzeroski, Raedt & Blockeel, 1998) has aroused interest as an efficient approach for learning policies for relational domains. RRL includes a set of learning techniques for computing the optimal policy for reaching the given goals by exploring the state space though trial and error. The major benefit of these techniques is that they can be used to solve problems whether the action model is known or not. In the other hand, since RRL does not explicitly include the task goals in the policies, new policies have to be learned every time a new goal has to be achieved, even if the dynamics of the environment has not changed.

In general, domain-specific planners have to deal with the problem of generalization. These techniques build planning programs from a given set of solved problems so cannot theoretically guarantee solving subsequent problems.

Learning Domain Models

No matter how efficient a planner is, if it is fed with a defective domain model, it will return defective plans. Designing, encoding and maintaining a domain model is very laborious. At the time being, planners are the only tool available to assist in the development of an AP domain model, but planners are not designed specifically for this purpose. Domain model learning studies ML mechanisms to automatically acquire the planning action schemas (the action preconditions and post-conditions) from observations of action executions.

Learning domain models in deterministic environments is a well-studied problem; diverse inductive learning techniques have been successfully applied to automatically define the actions schema from observations (Shen & Simon, 1989), (Benson, 1997), (Yang, Wu & Jiang, 2005), (Shahaf & Amir, 2006). In stochastic environments, this problem becomes more complex. Actions may result in innumerable different outcomes, so more elaborated approaches are required. (Pasula, Zettlemoyer & Kaelbling, 2004) presented the first specific algorithm to learn simple stochastic actions without conditional effects. This algorithm is based on three levels of learning: the first one consists of deterministic rule-learning techniques to induce the action preconditions. The second one relies on a search for the set of action outcomes that best fits the execution examples, and; the third one consists of estimating the probability distributions over the set of action outcomes. But, stochastic planning algorithms do not need to consider all the possible actions outcomes. (Jimenez & Cussens 2006) proposed to learn complex action-effect models (including conditions) for only the relevant action outcomes. Thus, planners generate robust plans by covering only the most likely execution outcome while leaving others to be completed when more information is available.

In deterministic environments, (Shahaf & Amir, 2006) introduced an algorithm that exactly learns STRIPS action schemas even if the domain is only partially observable. But, in stochastic environments, there is still no general efficient approach to learn action model.

FUTURE TRENDS

Since the appearance of the first PDDL version in IPC-1998, the standard planning representation language has evolved to bring together AP algorithms and real-world planning problems. Nowadays, the PDDL 3.0 version for the IPC-2006 includes numeric state variables to support quality metrics, durative actions that allow explicit time representation, derived predicates to enrich the descriptions of the system states, and soft goals and trajectory constraints to express user preferences about the different possible plans without discarding valid plans. But, most of these new features are not handled by the state-of-the-art planning algorithms: The existing planners usually fail solving problems that define quality metrics. The issue of goal and trajectory preferences has only been initially addressed. Time and resources add such extra complexity to the search process that a real-world problem becomes extremely difficult to solve. New challenges for the AP community are those related to developing new planning algorithms and heuristics to deal with these kinds of problems. As

it is very difficult to find an efficient general solution, ML must play an important role in addressing these new challenges because it can be used to alleviate the complexity of the search process by exploiting regularity in the space of common problems.

Besides, the state-of-the-art planning algorithms need a detailed domain description to efficiently solve the AP task, but new applications like controlling underwater autonomous vehicles, Mars rovers, etc. imply planning in environments where the dynamics model may be not easily accessible. There is a current need for planning systems to be able to acquire information of their execution environment. Future planning systems have to include frameworks that allow the integration of the planning and execution processes together with domain modeling techniques.

Traditionally, learning-based planners are evaluated only against the same planner but without learning, in order to prove their performance improvement. Additionally, these systems are not exhaustively evaluated; typically the evaluation only focuses on a very small number of domains, so these planners are usually quite fragile when encountering new domains. Therefore, the community needs a formal methodology to validate the performance of the new learning-based planning systems, including mechanisms to compare different learning-based planners.

Although ML techniques improve planning systems, existing research cannot theoretically demonstrate that they will be useful in new benchmark domains. Moreover, for time being, it is not possible to formally explain the underlying meaning of the learned knowledge (i.e., does the acquired knowledge subsumes task decomposition? a goal ordering? a solution path?). This point reveals that future research in AP and ML will also focus on theoretical aspects that address these issues.

CONCLUSION

Generic domain-independent planners are still not able to address the complexity of real planning problems. Thus, most planning systems implemented in applications require additional knowledge to solve the real planning tasks. However, the extraction and compilation of this specific knowledge by hand is complicated.

This article has described the main last advances in developing planners successfully assisted by ML

techniques. Automatic learned knowledge is useful for AP in diverse ways: it helps planners in guiding search processes, in completing domain theories or in specifying particular solutions to a particular problem. However, the learning-based planning community can not only focus on developing new learning techniques but also on defining formal mechanisms to validate its performance against other generic planners and against other learning-based planners.

REFERENCES

Benson, S. (1997). Learning Action Models for Reactive Autonomous Agents. PhD thesis, Stanford University.

Blum, A., & Furst, M. (1995). Fast planning through planning graph analysis. In C. S. Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI-95*, volume 2, pages 1636–1642, Montreal, Canada, August 1995. Morgan Kaufmann.

Bonet, B. & Geffner, H. (2001). Planning as Heuristic Search. *Artificial Intelligence*, 129 (1-2), 5-33.

Borrajo, D., & Veloso, M. (1997). Lazy Incremental Learning of Control Knowledge for Efficiently Obtaining Quality Plans. *AI Review Journal. Special Issue on Lazy Learning.* 11 (1-5), 371-405.

Botea, A., Enzenberger, M., Müller, M. & Schaeffer, J. (2005). Macro-FF: Improving AI Planning with Automatically Learned Macro-Operators. *Journal of Artificial Intelligence Research (JAIR)*, 24, 581-621.

Bylander, T., The computational complexity of propositional STRIPS planning. (1994). *Artificial Intelligence*, 69(1-2), 165–204.

Coles, A., & Smith, A. (2007). Marvin: A heuristic search planner with online macro-action learning. *Journal of Artificial Intelligence Research*, 28, 119–156.

De la Rosa, T., García Olaya, A., & Borrajo, D. (2007) Using Utility Cases for Heuristic Planning Improvement. *Procceedings of the 7th International Conference on Case-Based Reasoning*, Belfast, Northern Ireland, Springer-Verlag.

Dzeroski, S., Raedt, L. D., & Blockeel, H., (1998) Relational reinforcement learning. In *International*

Workshop on Inductive Logic Programming, pages 11–22.

Fikes, R., Hart, P., & Nilsson, N., (1972) Learning and Executing Generalized Robot Plans, *Artificial Intelligence*, 3, pages 251-288.

Fox, M. & Long, D, (2003) PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20, 61–124.

Hoffmann J. & Nebel B. (2001) The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*,14, 253–302.

Jiménez, S. & Cussens, J. (2006). Combining ILP and Parameter Estimation to Plan Robustly in Probabilistic Domains. *In Conference on Inductive Logic Programming. Santiago de Compostela, ILP2006.* Spain.

Khardon, R. (1999) Learning action strategies for planning domains. *Artificial Intelligence*, 113, 125–148,

Pasula, H. Zettlemoyer, L. & Kaelbling, L. (2004) Learning probabilistic relational planning rules. *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling, ICAPS04.*

Samuel, A. L., (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 211–229.

Shahaf, D & Amir, E. (2006). Learning partially observable action schemas. *In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06).*

Shen, W. & Simon. (1989). Rule creation and rule learning through environmental exploration. In *Proceedings of the IJCAI-89*, pages 675–680.

Spector, L. (1994) Genetic programming and AI planning systems. In *Proceedings of Twelfth National Conference on Artificial Intelligence*, Seattle, Washington, USA, AAAI Press/MIT Press.

Veloso, M. (1994). *Planning and learning by analogical reasoning*. Springer Verlag.

Winner, E. & Veloso, M. (2003) Distill: Towards learning domain-specific planners by example. In Proceedings of Twentieth International Conference on Machine Learning (ICML 03), Washington, DC, USA.

Yang, Q, Wu, K & Jiang, Y. (2005) Learning action models from plan examples with incomplete knowledge. *In Proceedings of the 2005 International Conference on Automated Planning and Scheduling, (ICAPS 2005) Monterey, CA USA*, pages 241–250.

Yoon, S., Fern, A., & Givan, R., (2006). Learning heuristic functions from relaxed plans. *In International Conference on Automated Planning and Scheduling (ICAPS-2006)*.

KEY TERMS

Control Rule: *IF-THEN* rule to guide the planning search-tree exploration.

Derived Predicate: Predicate used to enrich the description of the states that is not affected by any of the domain actions. Instead, the predicate truth values are derived by a set of rules of the form **if** formula(x) **then** predicate(x).

Domain Independent Planner: Planning system that addresses problems without specific knowledge of the domain, as opposed to *domain-dependent planners,* which use domain-specific knowledge.

Macro-Action: Planning action resulting from combining the actions that are frequently used together in a given domain. Used as control knowledge to speed up plan generation.

Online Learning: Knowledge acquisition during a problem-solving process with the aim of improving the rest of the process.

Plateau: Portion of a planning search tree where the heuristic value of nodes is constant or does not improve.

Policy: Mapping between the world states and the preferred action to be executed in order to achieve a given set of goals.

Search Control Knowledge: Additional knowledge introduced to the planner with the aim of simplifying the search process, mainly by pruning unexplored portions of the search space or by ordering the nodes for exploration.

A Longitudinal Analysis of Labour Market Data with SOM

Patrick Rousset
CEREQ, France

Jean-Francois Giret
CEREQ, France

INTRODUCTION

The aim of this paper is to present a typology of career paths in France drawn up with the Kohonen algorithm and its extension to a clustering method of life history analysis based on the use of Self Organizing Maps (SOMs). Several methods have previously been presented for transforming qualitative into quantitative information so as to be able to apply clustering algorithms such as SOMs based on the Euclidean distance. Our approach consists in performing quantitative encoding on labor market situation proximities across time. Using SOMs, the preservation of the topology also makes it possible to check whether this new method of encoding preserves the particularities of the life history according to our economic approach to careers. Lastly, this quantitative encoding preprocessing, which can be easily applied to analysis methods of life history, completes the set of methods extending the use of SOM to qualitative data.

BACKGROUND

Several methods are generally used to study the dynamic aspects of careers. The first method, which estimates some reduced-form transition models, has been extensively used in labor microeconometrics, using event-history models for continuous-time data or discrete-time panel data with Markov processes. Those of the second kind, which include the method presented here, are sequence analysis methods dealing with complex information about individual labor market histories, such as the various states undergone, the duration of the spells, multiple transitions between the states, etc.. The idea was to empirically generate a statistical typology of sequences by performing cluster analysis (Lebart, 2006). This method thus makes it pos-

sible to define "cluster paths" constituting endogenous variables and explained in terms of individual characteristics such as gender, educational level or parental socio-economic status. The optimal matching method, which has been widely used in social science since the pioneering paper by Abott (Abbott & Hrycak, 1990) , is an attractive solution for analysing longitudinal data of this kind. The basic idea underlying this method is to take a pair of sequences and calculate the cost of transforming them into each other by performing a series of elementary operations (insertion, deletion and substitution). However, this method has been heavily criticized because it may be difficult to determine the values of these elementary operations. Here we adopt another strategy. First, in order to classify sequences into groups, we have defined a measure of the distance between each trajectory, which is coherent with our data and with some well-known theoretical hypotheses in the field of labor economics. We then use Self Organizing Maps (the Kohonen algorithm) for classification and purposes.

Self Organizing Maps (see Kohonen, 2001, Fort, 2006) are known to be a powerful clustering and projection method. Since this method accounts efficiently for changes occurring with time, SOMs yield accurate predictions (see for example Cotrell , Girard & Rousset ,1998, Dablemont, Simon, Lendasse, Ruttiens, Blayo & Verleysen, 2003, Souza, Barreto & Mota, 2005). Life histories can be considered as a qualitative record of information, while SOMs are based on Euclidean distance. Many attempts have been made to transform qualitative variables into quantitative ones: using for example the Burt description (see the KACM presentation in Cottrel & Letremy, 1995) or using the multidimensional scaling (Miret, Garcia-Lagos, Joya, Arazoza & Sandoval, 2005). In our approach, the quantitative recoding focuses on the proximity between items considering particularities of the data (a life his-

tory) according to our economic approach. When the preprocessing of recoding is performed, Self Organizing Maps is a useful clustering tool, first considering its pre-mentioned clustering and projection qualities and also because of its ability to make the efficiency of our new encode emerge.

CLUSTERING LIFE HISTORY WITH SOM

An Example of a Life History

Career Paths

Labor economists have generally assumed that the beginning of a career results from a matching process (Jovanovich, 1979). Employers and job seekers lack information about each other: employers need to know how productive their potential employee is and job applicants want to know whether the characteristics of the job correspond to their expectations. Job turnover and temporary employment contracts can therefore be viewed as the consequences of this trial-by-error process. However, individuals' first employment situations may also act as a signal of employability to the labor market. For example, a long spell of unemployment during the first years in a person's career may be interpreted by potential employers as sign of low work efficiency; whereas working at a temp agency may be regarded as a sign of motivation and adaptability. This is consistent with the following path dependency hypothesis: the influence of past job experience on the subsequent career depends on the "cost" associated with the change of occupational situation. However, empirical studies have shown that employers mainly recruit on the basis of recent work experience (Allaire, Cahuzac & Tahar, 2000). The effects of less recent employment situations on a person's career therefore decrease over time.

Data

The data used in this study were based on the "Generation 98" survey carried out by Céreq: 22 000 young people who had left initial training in 1998 at all levels and in all training specializations were interviewed in spring 2001 and 2003 and autumn 2005. This sample

was representative of the 750 000 young people leaving the education system for the first time that year in France. This survey provided useful information about the young people's characteristics (their family's socio-economic status, age, highest grade completed, highest grade attended, discipline, any jobs taken during their studies, work placement) and the month-by-month work history from 1998 to 2005. We therefore have a complete and detailed record of the labor market status of the respondents during the 88-month period from July 1998 to November 2005. Employment spells were coded as follows, depending on the nature of the labor contract: 1 = permanent labor contract, 2 = fixed term contract, 3 = apprenticeship contract, 4 = public temporary labor contract, 5 = interim/temping). Other unemployed situations were coded as follows: 6 = unemployment, 7 = inactivity, 8 = military service, 9 = at school.

Preprocessing Phase: Life History Encoding

The encoding of the trajectories involved a two-step preprocessing phase : defining a distance between states including time dynamics and the resulting quantitative encoding of trajectories. These two steps refer to the specificity of the data set structures of life history samples: the variables items (the states) are some qualitative information while the variables order records some quantitative information (the timing and the duration of events).

The Distance Between Situations

Working with pairs (state, time), called situations, allows to include the time dynamics in the proximities between occupational states. The proximity between two situations is measured on the basis of their common future, in line with our Economic approach. A situation is assumed as a potential for its own future, depending on its influence on this future. The similarity between two situations is deduced from comparisons between their referring potential. The *potential future* P^S of a situation S among n monthly periods and p states is defined as the $p \times n$ dimensional vector given in (1). Its components $P_S^{S'}$ are the product of terms φ and β. φ measures the flow between situation S and any situation S' as the empirical probability of reaching S' starting

from S. It is also the empirical probability of an individual i being in any future situation S' conditionally of being at the present in S. The coefficient of temporal inertia β weights the influence of S' on P^S according to the Economic approach. It is a decreasing function of the time delay (t'-t). In the career paths application, the function chosen is the inverse of the delay and 0 for the past. Lastly, α ensures that potential futures P_S will be profiles. The natural distance between situations is therefore the χ^2 distance between their potential future profiles.

$$P_{S=(s,t)} = \frac{1}{\alpha(t)}(\mathbf{0},...,\mathbf{0}, \underbrace{\beta(t')\Phi_{S=(s,t)}^{S'=(1,t')}, \beta(t')\Phi_S^{S'=(2,t')},...,\beta(t')\Phi_S^{S'=(9,t')}}_{t'\geq t},...,$$

$$\underbrace{\beta(T)\Phi_S^{S'=(1,T)},...,\beta(T)\Phi_S^{S'},...,\beta(T)\Phi_S^{S'=(9,T)}}_{T})$$

(1)

where

$$\Phi_{S=(s,t)}^{S'=(s',t')} = \frac{\sum\limits_i \chi_S \chi_{S'}}{\sum\limits_i \chi_S}{}_{,}^{1}$$

$$\beta(t,t') = \frac{1}{t'-t+1}$$

and

$$\alpha^{-1}(t) = \sum_{S'}\beta(S')\Phi_S^S$$

.

The Trajectory Encoding

In the present case of equi-weighting, the inertia of the space of situations results from the distances previously computed. The principal components of inertia, called here principal events, can therefore be deduced. The term "event" refers to a combination of the point in time, the duration and the occupational status. The quantitative encoding of trajectories proposed here results from their description in the "events" space.

The process used here is in line with J.P. Benzécri's one (Benzécri, 1973), which explains how: when considering a set of situations $\{S^i\}$, its center of gravity G and the matrix recording squares of distance ($d^{ij'}$) between elements S^j and $S^{j'}$, one can deduce the matrix of scalar products Δ between any vectors GS^j

with the formula (2). Applying to the situations, the principal components of inertia (the principal events) are computed as the principal component vectors of the matrix Δ.

Trajectories can then be described in the principal events space: performing the traditional binary encoding (3) of the trajectory T_i is equivalent to performing a linear encoding through the situations (4) and then also through the principal events E (5).

$$\Delta_j^{j'} = GS^j.GS^{j'} = -\frac{1}{2}\left[d^{jj'} - d^{j\cdot} - d^{\cdot j'} + d^{\cdot\cdot}\right]$$

(2)

$$T_i = (\underbrace{0,...,0,1,0,...,0,0}_{time=1},\underbrace{...,0,1,0,...,0}_{9\ positions},...,\underbrace{0,...,0,1,0,...,0}_{time=9})$$

(3)

$$T_i = \sum_{S=(s,t)} \alpha_{(S)}(0,...,\underset{\underset{position\ s*t}{\uparrow}}{0,1,0},...,0)$$

(4)

$$T_i = \sum_S \alpha_S (\sum_e \beta_e^S E_e) = \sum_e \gamma_e E_e$$

(5)

CLASSIFICATION OF LIFE HISTORIES WITH SOM: A TYPOLOGY OF CAREER PATHS IN FRANCE AND ECONOMIC INTERPRETATIONS

The result of the typology of career paths in France using Self Organizing Maps with a 10x10 grid is presented in Figure 1. In each unit of the map, a chronogram describes the characteristics of the class (the career path). Chronograms show the evolution in time of the proportional contribution (in percentage) of each state to the classes. On the one hand, the SOM topology reflects the continuity of the evolution in time. On the other hand, similarities between situations give rise to the mixing of classes (see for example cluster 95) or proximities on the map between two clusters (for example, between clusters 71 and 72 – eighth line, first and second column) although few individuals are in the same state at the same time. The map thus makes it possible to assess the efficiency of the encoding process.

The Kohonen map displays a concise vision of the types of career paths occurring during the first seven years of working life. In general, most of the clusters describe a direct school-to-work transition process,

1031

Figure 1. Typology of career paths with SOM: each unit on the map gives a chronogram of the evolution in time of the proportional contribution (as a percentage) of any occupational position. Two populations of closed units have similar chronogram.

which can be characterized by an immediate access to a permanent contract or an indirect access during the first few years. In the upper-left-hand corner of the map, mainly the five clusters of the two first lines, career paths are characterized by a high level of access to employment with permanent contracts. Young people rapidly gained access to a permanent contract and kept it until the end of the observation period. In upper-right-hand corner of the map, access to a permanent contract was less direct: during the first year, young people first obtained a temporary contract or spent ten months in compulsory military service before obtaining a permanent contract. the upper part of the map, the bottom part describes career paths with longer period of temporary contracts and/or unemployment. In the lower-right-hand corner, access to a permanent contract is becoming rare during the first few years. However, more than ninety per cent of young people have obtained a final permanent position (in the last column of the map). In the bottom lines, a five-year public policy contract called "Emploi Jeunes" features

strongly instead of the classical fixed term contract. The lower-left-hand corner of the map shows more unstable trajectories which end in a temporary position: seven years after leaving school, people have a temporary contract (in the last two clusters in the first column) or are unemployed (second and third cells on the last line). The chronograms situated in the middle-left-hand part of the map highlight how the longitudinal approach is interesting to understand the complexity of transition processes: the young people here were directly recruited for a permanent job, but five or six years after graduating, they lost this permanent job. This turn of events can be explained by the strong change in the economic environment which occurred during this period; the years 2003 and 2004 correspond to a dramatic growth of youth unemployment on the French labor market.

What role does each individual characteristic play in the development of these career paths? Several factors may explain the labor market opportunities of school-leavers: human capital factors, parents' social class,

Figure 2. Career path typology by educational levels
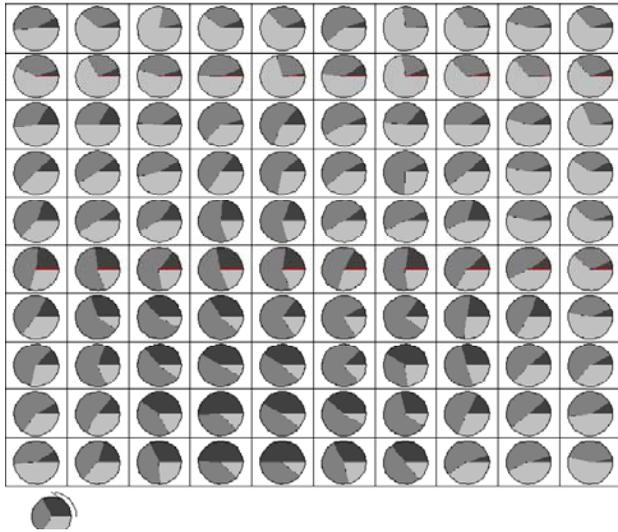
Educational levels (dark = without qualifications; medium = secondary educational graduates; light = ligher educational graduates).

and other factors responsible for inequalities on the labor market, such as parents' nationality and gender. The distribution of these characteristics was included graphically in each cell on the map. Figure 2, which gives the distribution in terms of educational level, clearly shows that educational level strongly affected the career path. Higher educational graduates feature much more frequently in the upperleft-hand corner of the map, whereas school leavers without any qualifications occur more frequently in the bottom part. Figure 3 shows a similar pattern as far as gender is concerned: there are much higher percentages of females than males in the most problematic career paths, which suggests the occurrence of gender segregation or discriminatory practices on the French labor market. The differences are less conspicuous as far as the father's nationality is concerned (Figure 4). However, the results obtained here also suggests that children with French parents have better chances than the others of finding "safe" career paths.

Figure 3. Career path typology by gender
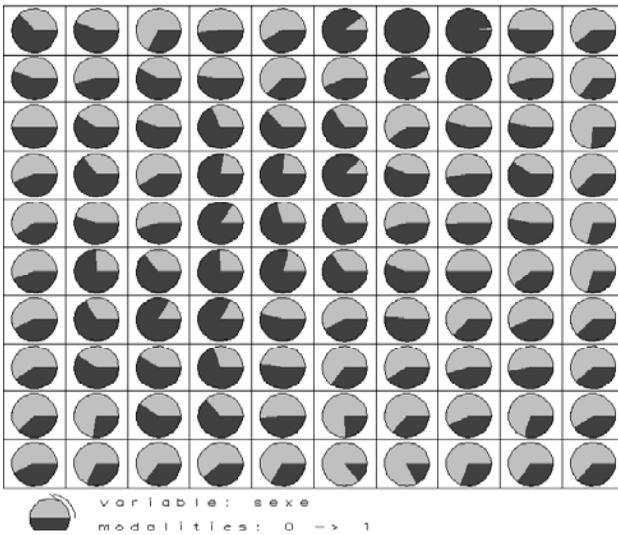
Dark =male, light = female

Figure 4. Career path typology by father's nationality
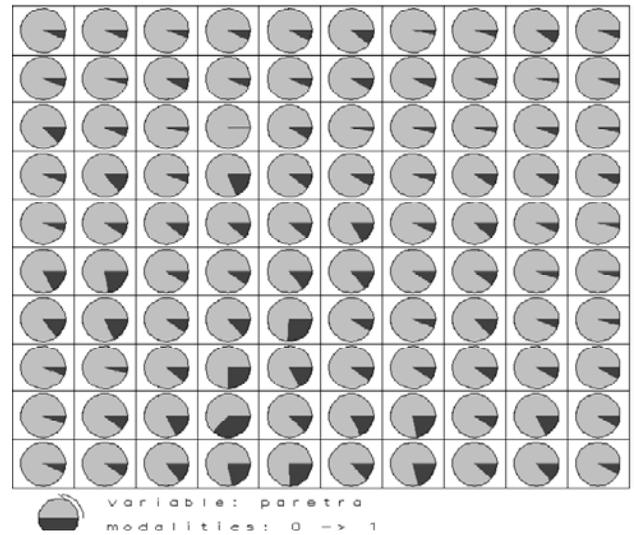
Father's nationality: dark = foreign nationality, light = French nationality

FUTURE TRENDS

The relevance of the method presented here concerns both aspects: the preprocessing and SOMs' result. The advantages of SOM depend on the distance chosen, which must enable the associated algorithm to preserve the proximities between situations. The relevance of the preprocessing stage in the method will therefore be confirmed if it enhances the reliability of the SOMs (Debodt, Cottrell & Verleysen, 2002, and Rousset, Guinot & Maillet, 2006, have presented a method of measuring and a method of increase reliability, respectively). On the other hand, the reliability also depends on the choice of the future weighting function, function β in formula (1). By consequence, function β could be determined here with a view to the reliability of the SOM results. But unfortunately, in general, this approach may be counter-productive in some cases. In the case of career paths, for example, it would lead to weighting the long term future, which would increase the robustness but would not be suitable from the point of view of the Economic investigation. This problem arises in many general contexts where the main effects of the present on the future are short term effects, whereas the reliability increases in the long term. The main criterion used to choose function β must therefore be the topic of interest.

Further studies are now required to improve the reliability of this method: first function β needs to be defined more closely and secondly, the validity of the method needs to be tested after enhancing the reliability that of the SOM topology obtained after performing the preprocessing step described above). It might also be worth investigating the use of Markovian models to define function β in particular, as well as to study career paths in general. This method will also have to be applied in the future to other samples.

CONCLUSION

The aim of this study was to analyze the early career of French schools leavers using Self Organizing Maps. This empirical analysis showed that career paths are strongly segmented. Although most of the "career paths" studied were characterized by stabilization on the labor market at some point or another, some of them show the great difficulties encountered by labor market entrants. Obtaining a permanent contract does not actually guarantee life-long employment. In addition, the econometric analysis carried out in the second part of this study shows that the diversity of career paths can be partly explained by the educational levels and individual characteristics of school leavers.

In the present method of analyzing information on individuals' trajectories in time through a finite number of states, two important aspects are combined: the encoding of the data and the analysis of the data presented in the form of SOMs. The first aspect avoids the well known problem of skew present with qualitative encoding, including when it is linked to the evolution in time. Self organizing maps are a natural approach to the data analysis, since this tool combines the advanteges of clustering and representation methods. The method described here turns out to be an efficient means of investigating changes with time and the proximities between situations. In addition, the preservation of the topology was found to be a useful property, which makes it possible to assess the efficiency of the recoding. In conclusion, the method presented here could easily be used to analyze any life history.

REFERENCES

Abbott, A. & Hrycak, A. (1990). Measuring resemblance in sequence data: an optimal matching analysis of musicians' career. *American Journal of Sociology*, 96, 1, 144-185.

Allaire, G., Cahuzac, E. & Tahar, G. (2000). Persistance du chômage et insertion. *L'Actualité économique*, 76, 2, 237-263.

Benzécri J.P. (1973) L'Analyse des Données. TIIBn°2 "Representation Euclidienne d'un Ensemble fini de masses et de distances" , DUNOD, Paris, 65-95.

Cottrell, M., Girard, B. & Rousset, P. (1998). Forecasting of curves using a Kohonen classification, Journal of Forecasting, 17, 429-439.

Cottrel, M. & Letrémy, P. (1995). Classification et analyse des correspondances au moyen de L'Algoritme de Kohonen: Application à l'étude de données socio-économiques, *Prépublication du SAMOS*, (42), University of Paris I, France..

De Bodt, E., Cottrell, M., & Verleysen, M. (2002). Statistical tools to assess the reliability of self-organizing

maps. *Neural Networks*, *15*, 967-978.

Dablemont, S., Simon, G., Lendasse, A., Ruttiens, A., Blayo, F. & Verleysen, M. (2003). Time series forecasting with SOM and local non-linear models - Application to the DAX30 index prediction. In Proceedings of the 4th Workshop on Self-Organizing Maps, (WSOM)'03, 340-345.

Fort J.C. (2006). SOM's mathematics , *Neural Networks*, 19, 812-816.

Fougère, D. & Kamionka, T. (2005). Econometrics of Individual Labor Market Transitions, IZA Discussion Papers 1850, Institute for the Study of Labor (IZA).

Jovanovic, C. (1979). Job matching and the theory of turnover, *Journal of Political Economy*, 87, 5, 972-990.

Kohonen T. (2001). Self-Organizing Maps. 3.ed, *Springer Series in Information Sciences*, 30, Springer Verlag, Berlin.

Lebart L., Morineau M. et Piron M. (2006). *Statistique exploratoire multidimensionnelle*. 4.ed, Dunod, Paris.

Miret, E., García-Lagos, F., Joya, G., Arazoza, H. & Sandoval, F. (2005). A combined multidimensional scaling + self-organizing maps method for exploratory analysis of qualitative data. Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM'05), Paris, France, 711-718.

Rousset P., Guinot C. & Maillet B. (2006). Understanding and Reducing Variability of SOM neighbourhood structure , *Neural Networks*, 19, 838-846.

Souza, L. G., Barreto, G.A. & Mota, J. C. (2005). Using the self-organizing map to design efficient RBF models for nonlinear channel equalization, Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM'05), Paris, France, 339-346.

KEY TERMS

Careers Paths: Sequential monthly position among several pre-defined working categories.

Distibutional Equivalency: Property of a distance that allows to group two modalities of the same variable having identical profiles into a new modality weighted with the sum of the two weights.

Markov Process: Stochastic process in wich the new state of a system depends on the previous state or a finite set of previous states.

Optimal Matching: Statistical method issued from biology abble to compare two sequences from a predifined cost of substitution.

Preservation of Topology: After learning, observations associated to the same class or to « close » classes according to the definition of the neighborhood and given by the network structure are « close » according to the distance in the input space.

Self-Organizing Maps by Kohonen: A neural network unsupervised method of vector quantization widely used in classification. Self-Organizing Maps are a much appreciated for their topology preservation property and their associated data representation system. These two additive properties come from a pre-defined organization of the network that is at the same time a support for the topology learning and its representation.

χ² Distance: Distance having certain specific properties such as the distibutional equivalency.

Managing Uncertainties in Interactive Systems

Qiyang Chen
Montclair State University, USA

John Wang
Montclair State University, USA

INTRODUCTION

To adapt users' input and tasks an interactive system must be able to establish a set of assumptions about users' profiles and task characteristics, which is often referred as user models. However, to develop a user model an interactive system needs to analyze users' input and recognize the tasks and the ultimate goals users trying to achieve, which may involve a great deal of uncertainties. In this chapter the approaches for handling uncertainty are reviewed and analyzed. The purpose is to provide an analytical overview and perspective concerning the major methods that have been proposed to cope with uncertainties.

Approaches for Handling Uncertainties

For a long time, the Bayesian model has been the primary numerical approach for representation and inference with uncertainty. Several mathematical models that are different from the probability prospective have also been proposed. The main ones are Shafer-Dempster's Evidence Theory (Belief Function) (Shafer, 1976; Dempster, 1976) and Zadeh's Possibility Theory (Zadeh, 1984). There have also been some attempts to handle the problem of incomplete information using classical logic. Many approaches to default reasoning logic have been proposed, and study of non-monotonic logic has gained much attention. These approaches can be classified into two categories: numerical approaches and non-numerical approaches.

1. *Probability and Bayesian Theory.* There is support for the theoretical necessity and justification of using a probability framework for knowledge representation, evidence combination and propagation, learning ability, and clarity of explanation (Buchana and Smith, 1988). Bayesian processing remains the fundamental idea underlying many

new proposals that claim to handle uncertainty efficiently.

In all the practical developments to date, the Bayesian formula and probability values have been used as some kind of coefficients to augment deterministic knowledge represented by production rules (Barr and Feigenbaum, 1982). Some intuitive methods for combination and propagation of these values have been suggested and used. One such case is the use of Certainty Factors (CF) in MYCIN (Shortliffe and Buchanan, 1976). Rich also use a simplified CF approach in user modeling system GROUNDY (Rich, 1979).

However, some objections against such probabilistic methods of accounting for uncertainty have been raised (Karnal and Lemmer, 1986). One of the main objections is that these values lack any definite semantics because of the way they have been used. Using a single number to summarize uncertainty information has always been a contested issue (Heckerman, 1986).

The Bayesian approach requires that each piece of evidence be conditionally independent. It has been concluded that the assumptions of conditional independence of the evidence under the hypotheses are inconsistent with the other assumptions of exhaustive and mutually exclusive space of hypotheses. Specifically, Pednault *et al.* (1981) show that, under these assumptions, a probabilistic update could take place if there were more than two competing hypotheses. Pearl (1985) suggests that the assumption of conditional independence of the evidence under the negation of the hypotheses is over-restrictive. For example, if the inference process contains multiple paths linking the evidence to the same hypothesis, the independence is violated. Similarly, the required mutual exclusiveness and exhaustiveness of the hypotheses are not very realistic. This assumption would not hold if more than one hypothesis occurred simultaneously and is as restrictive as the single-default assumption of the simplest

diagnosing systems. This assumption also requires that every possible hypothesis is known *a priori*. It would be violated if the problem domain were not suitable to a closed-world assumption.

Perhaps the most restrictive limitation of the Byesian approach is its inability to represent ignorance. The Bayesian view of probability does not allow one to distinguish uncertainty from ignorance. One cannot tell whether a degree of belief was directly calculated from evidence or indirectly inferred from an absence of evidence. In addition, this method requires a large amount of data to determine the estimates for prior and conditional probabilities. Such a requirement becomes manageable only when the problem can be represented as a sparse Bayesian network that is formed by a hierarchy of small clusters of nodes. In this case, the dependencies among variables (nodes in the network) are known, and only the explicitly required conditional probabilities must be obtained (Pearl, 1988).

2. *The Dempster-Shafer Theory of Evidence.* The Dempster-Shafer theory, proposed by Shafer (Shafer, 1976), was developed within the framework of Dempster's work on upper and lower probabilities induced by a multi-valued mapping (Dempster, 1967). Like Bayesian theory, this theory relies on degrees of belief to represent uncertainty. However, it allows one to assign a degree of belief to subsets of hypotheses. According to the Dempster-Shafer theory, the feature of multi-valued mapping is the fundamental reason for the inability of applying the well-known theorem of probability that determines the probability density of the image of one-to-one mapping (Cohen, 1983). In this context, the lower probability is associated with the degree of belief and the upper probability with a degree of plausibility. This formalism defines certainty as a function that maps subsets of a proposition space on the [0,1] scale. The sets of partial beliefs are represented by mass distributions of a unit of belief across the space of propositions. These distributions are called the basic probability assignment. The total certainty over the space is 1. A non-zero BPA can be given to the entire proposition space to represent the degree of ignorance. The certainty of any proposition is then represented by the interval characterized by upper and lower probabilities.

Dempster's rule of combination normalizes the intersection of the bodies of evidence from the two sources by the amount of non-conflictive evidence between the sources.

This theory is attractive for several reasons. First, it builds on classical probability theory, thus inheriting much of its theoretical foundations. Second, it seems not to over-commit by not forcing precise statements of probabilities: its probabilities do not seem to provide more information than is really available. Third, it reflects the degree of ignorance of the probability estimate. Fourth, the Dempster-Shafer theory provides rules for combining probabilities and thus for propagating measures through the system. This also is one of the most controversial points since the propagation method is an extension of the multiplication rule for independent events. Because many applications involve dependent events, the rule might be inapplicable by classical statistical criteria. The tendency to assume that events are independent unless proven otherwise has stimulated a large proportion of the criticism of probability approaches. Dempster-Shafer theory suffers the same problem (Bhatnager and Kanal, 1986).

In addition, there are two problems with Dempster-Shafer approach. The first problem is computational complexity. In the general case, the evaluation of the degree of belief and upper probability requires exponential time in the cardinality of the hypothesis set. This complexity is caused by the need for enumerating all the subsets of a given set. The second problem in this approach results from the normalization process presented in both Dempster's and Shafer's work. Zadeh has argued that this normalization process can lead to incorrect and counter-intuitive results (Zadeh, 1984). By removing the conflicting parts of the evidence and normalizing the remaining parts, important information may be discarded rather than utilized adequately. Dubois and Prade (1985) have also shown that the normalization process in the rule of evidence combination creates a sensitivity problem, where assigning a zero value or a very small value to a basic probability assignment causes very different results.

Based on Dempster-Shafer theory, Garvey *et al.* (1982) proposed an approach called Evidential Reasoning that adopts the evidential interpretation of the degree of belief and upper probabilities. This approach defines the likelihood of a proposition as a subinterval of the

unit interval [0,1]. The lower bound of this interval is the degree of support of the proposition and the upper bound is its degree of plausibility. When distinct bodies of evidence must be pooled, this approach uses the same Dempster-Shafer techniques, requiring the same normalization process that was criticized by Zadeh (Zadeh, 1984).

3. *Fuzzy Sets and Possibility Theory.* The theory of possibility was proposed independently by Zadeh, as a development of fuzzy set theory, in order to handle vagueness inherent in some linguistic terms (Zadeh, 1978). For a given set of hypotheses, a possibility distribution may be defined in a way that is very similar to that of a probability distribution. However, there is a qualitative difference between the probability and possibility of an event. The difference is that a high degree of possibility does not imply a high degree of probability, nor does a low degree of probability imply a low degree of possibility. However, an impossible event must also be improbable. More formally, Zadeh defined the concept of a possibility distribution.

The concept of possibility theory has been built upon fuzzy set theory and is well suited for representing the imprecision of vague linguistic predicates. The vague predicate induces a fuzzy set and the corresponding possibility distribution. From a semantic point of view, the values restricted by a possibility distribution are more or less all the eligible values for a linguistic variable. This theory is completely feasible for every element of the universe of discourse.

4. *Theory of Endorsement.* A different approach to uncertainty representation was proposed by Cohen (Cohen, 1983), which is based on a qualitative theory of "endorsement." According to Cohen, the records of the factors relating to one's certainty are called endorsements. Cohen's model of endorsement is based on the explicit recording of the justifications for a statement, normally requiring a complex data structure of information about the source. Therefore, this approach maintains the uncertainty. The justification is classified according to the type of evidence for a proposition, the possible actions required to solve the uncertainty of that evidence, and other related features.

Endorsements can provide a good mechanism for the explanations of reasoning, since they create and maintain the entire history of justifications (*i.e.,* reasons for believing or disbelieving a proposition) and the relevance of any proposition with respect to a given goal. Endorsements are divided into five classes: rules, data, task, conclusion, and resolution. Cohen points out that the main difference between the numerical approaches and the endorsement-based approach, specifically with respect to chains of inferences, is that reasoning in the former approach is entirely automatic and non-reflective, while the latter approach provides more information for reasoning about uncertainty. Consequently, reasoning in the latter approach can be controlled and determined by the quality and availability of evidence.

Endorsements provide the information necessary to many aspects of reasoning about uncertainty. Endorsements are used to schedule sure tasks before unsure ones, to screen tasks before activating them, to determine whether a proposition is certain enough for some purpose, and to suggest new tasks when old ones fail to cope with uncertainty. Endorsements distinguish different kinds of uncertainty, and tailor reasoning to what is known about uncertainty. However, Bonissone and Tong (1995) argue that combinations of endorsements in a premise (*i.e.*, proposition), propagation of endorsements to a conclusion, and ranking of endorsements must be explicitly specified for each particular context. This creates potential combinatorial problems.

5. *Assumption based reasoning and non-monotonic logic*: In the reasoned-assumptions approach proposed by Doyle (1979), the uncertainty embedded in an implication rule is removed by listing all the exceptions to that rule. When this is not possible, assumptions are used to show the *typicality* of a value (*i.e.*, default values) and defeasibility of a rule (*i.e.,* liability to defeat of reason). In classical logic, if a proposition C can be derived from a set of propositions S, and if S is a subset of T, then C can also be derived from T. As a system's premises increase, its possible conclusions at least remain constant and more likely increase. Deductive systems with this property are called monotonic. This kind of logic lacks tools for describing how to revise a formal theory to deal with inconsistencies caused by new information. McDermott and Dole proposed a non-monotonic

logic to cope with this problem (McDemott and Doyle, 1980).

When an assumption used in the deductive process is found to be false, non-monotonic mechanisms must be used to keep the integrity of the statements (Doyle, 1979). However, this approach lacks facilities for computing degrees of belief. Bonissone and Tong (1995) suggest that assumption-based systems can cope with cases of incomplete information, but they are inadequate in handling the imprecise information. In particular, they cannot integrate probabilistic measures with reasoned assumptions. Furthermore, such systems rely on the precision of the defaulted values. On the other hand, when specific information is missing, the system should be able to use analogous or relevant information inherited from some higher-level concept. This surrogate for the missing information is generally fuzzy or imprecise and provides limited constraints on the value of the missing information.

In the inference system employing non-monotonic logic, assumptions are made that may have to be revised in the light of new information. They have the property that at any given inference stage, more than one mutually consistent set of conclusions can be derived from the available data and possible assumptions. Such conclusions may be invalidated as new data is considered to be incompatible with some default assumptions. The inference system requires that justifications for any conclusion are recorded during the inference process and used for dependency-directed backtracking during the revision of beliefs. This is implemented by the Truth Maintenance System (TMS) (Doley, 1979).

The weakness of non-monotonic logic is that in standard non-monotonic logic the only message conveyed by a contradiction is that a piece of information previously believed true is actually false (for the time being). However, the real contents of the inconsistency being discovered may not be as reliable as was assumed, or it may be that a subject is not in a well-ordered state, or a mixture of both (Bahatnagar and Kanal, 1986). In addition, since the TMS examines the new information one piece at a time, it lacks the ability to detect *noise input* that should be ignored. This weakness is crucial to the task of pattern recognition (Chen and Norcio, 1997).

CONCLUSION

This chapter analyzes approaches of handling them in an adaptive human computer interface. Each approach can only deal with a particular type of uncertainty problems effectively. The interface system needs more comprehensive approach for uncertainty management due to various sources of uncertainties in human machine dialog. Especially, since human-machine dialog tend to be context-dependent, the management of uncertainty must provide a pattern-formatted view for user modeling. In other word, a user modeling system must examine the user input based the context of the dialog to obtain a complete and consistent user profiles. . Some non-traditional approaches have been proposed to handle uncertainties in interactive systems, such as neural networks and generic algorithms (Chen and Norcio 1997), because they have strong ability of pattern recognition and classification. However, the conversion between non-numerical user input and numerical input for neural network processing still involves a great deal of uncertainties.

REFERENCES

Barr, A. and Feigenbaum, E. A., *The Handbook of Artificial Intelligence 2*. Los Altos, Kaufmann , 1982.

Bhatnager, R. K. and Kanal, L. N., "Handling Uncertainty Information: A Review of Numeric and Nonnumeric Methods," *Uncertainty in Artificial Intelligence*, Kanal, L. N. and Lemmer, J. F. (ed.), pp2-26, 1986.

Bonissone, P. and Tong, R. M., "Editorial: Reasoning with Uncertainty in Expert Systems," *International Journal of Man-Machine Studies*, Vol. 30, 69-111 (2005)

Buchanan, B. and Smith, R. G. Fundamentals of Expert Systems, *Ann. Rev., Computer Science*, Vol. 3, pp. 23-58, 1988.

Chen, Q. and Norcio, A.F. "Modeling a User's Domain Knowledge with Neural Networks," *International Journal of Human-Computer Interaction*, Vol. 9, No. 1, pp. 25-40, 1997.

Chen, Q. and Norcio, A.F. "Knowledge Engineering in Adaptive Interface and User Modeling," *Human-*

Computer Interaction: Issues and Challenges, (ed.) Chen, Q. Idea Group Pub. 2001.

Cohen, P. R. and Grinberg, M. R., " A Theory of Heuristic Reasoning about Uncertainty, AI Magazine, Vol. 4(2), pp. 17-23, 1983.

Dempster, A. P., "Upper and Lower Probabilities Induced by a Multivalued mapping," *The Annuals of Mathematical Statistics* , Vol. 38(2), pp. 325-339, 1967.

Dubois, D. and Prade, H., "Combination and Propagation of Uncertainty with Belief Functions -- A Reexamination," *Proc. of 9th International Joint Conference on Artificial Intelligence,* pp. 111-113, 1985.

Dutta, A., "Reasoning with Imprecise Knowledge in Expert Systems," Information Sciences, Vol. 37, pp. 2-24, 2005.

Doyle, J., "A Truth Maintenance System," *AI,* Vol. 12, 1979, pp. 231-272.

Garvey, T. D., Lowrance, J. D. and Fischer, M. A. "An Inference Technique for Integrating Knowledge from Disparate Source," *Proc. of the 7th International Joint Conference on AI,* Vancouver, B. C. pp. 319-325, 1982

Heckerman, D., "Probabilistic Interpretations for MYCIN's Certainty actors," *Uncertainty in Artificial Intelligence,*(ed.). Kanal, L. N. and Lemmer, J. F. , 1986

Jacobson, C. and Freiling, M. J. "ASTEK: A Mulitparadigm Knowledge Acquisition tool for complex structured Knowledge." *International. Journal of Man-Machine Studies,* Vol. 29, 311-327. 1988.

Kahneman, D. and Tversky, A (1982). Variants of Uncertainty, Cognition, 11, 143-157.

McDermott, D. and Doyle, J., "Non-monotonic Logic," *AI* Vol. 13, pp. 41-72. (1980).

Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publisher, San Mateo, CA ,1988.

Pearl, J., "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach," *Proc. of the 2nd National Conference on Artificial Intelligence*, IEEE Computer Society, pp. 1-12, 1985.

Pednault, E. P. D., Zucker, S. W. and Muresan, L.V., "On the Independence Assumption Underlying Subjective Bayesian Updating, " *Artificial Intelligence*, 16, pp. 213-222. 1981

Raisinghani, M., Klassen, C. and Schkade, L. "Intelligent Software agents in Electonic Commerce: A Socio-Technical Perspective," *Human-Computer Interaction: Issues and Challenges,* (ed.) Chen, Q. Idea Group Pub. 2001.

Reiter, R., "A Logic for Default Reasoning," *Artificial Intelligence*, Vol. 13, 1980 pp. 81-132.

Rich, E., "User Modeling via Stereotypes," *Cognitive Sciences,* Vol. 3 1979, pp. 329-354.

Shafer, G., *A Mathematical Theory of Evidence,* Priceton University Press, 1976.

Zadeh, L. A., " Review of Books : A Mathematical Theory of Evidence," *AI Magazine.*, 5(3), 81-83. 1984

Zadeh, L. A. "Knowledge Representation in Fuzzy Logic," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 1, No. 1, pp. 89-100, 1989.

Zwick, R., "Combining Stochastic Uncertainty and Linguistic Inexactness: Theory and Experimental Evaluation of Four Fuzzy Probability Models," *Int. J. Man-Machine Studies,* Vol. 30, pp. 69-111, 1999.

KEY TERMS

Bayesian Theory: Also known as Bayes' rule or Bayes' law. It is a result in probability theory, which relates the conditional and marginal probability distributions of random variables. In some interpretations of probability, Bayes' theory tells how to update or revise beliefs in light of new evidences.

Default Reasoning: A non-monotonic logic proposed by Raymond Reiter to formalize reasoning with default assumptions. Default reasoningc can express facts like "by default, something is true"; by contrast, standard logic can only express that something is true or that something is false. This is a problem because reasoning often involves facts that are true in the majority of cases but not always. A classical example is: "birds typically fly". This rule can be expressed in

standard logic either by "all birds fly", which is inconsistent with the fact that penguins do not fly, or by "all birds that are not penguins and not ostriches and ... fly", which requires all exceptions to the rule to be specified. Default logic aims at formalizing inference rules like this one without explicitly mentioning all their exceptions

Non-Monotonic Logic: A formal logic whose consequence relation is not monotonic. Most studied formal logics have a monotonic consequence relation, meaning that adding a formula to a theory never produces a reduction of its set of consequences. Intuitively, monotonicity indicates that learning a new piece of knowledge cannot reduce the set of what is known. A monotonic logic cannot handle various reasoning tasks such as reasoning by default.

Possibility Theory: A mathematical theory for dealing with certain types of uncertainty and is an alternative to probability theory. Professor Lotfi Zadeh first introduced possibility theory in 1978 as an extension of his theory of fuzzy sets and fuzzy logic.

Shafer-Dempster's Evidence Theory: A mathematical theory of evidence based on belief functions and plausible reasoning, which is used to combine separate pieces of information (evidence) to calculate the probability of an event. The theory was developed by Arthur P. Dempster and Glenn Shafer.

Theory of Endorsement: An approach to represent uncertainty proposed by Cohen, which is based on a qualitative theory of "endorsement." According to Cohen, the records of the factors relating to one's certainty are called endorsements. Cohen's model of endorsement is based on the explicit recording of the justifications for a statement, normally requiring a complex data structure of information about the source. Therefore, this approach maintains the uncertainty. The justification is classified according to the type of evidence for a proposition, the possible actions required to solve the uncertainty of that evidence, and other related features.

Truth Maintenance System: A knowledge representation method for representing both beliefs and their dependencies. The name truth maintenance is due to the ability of these systems to restore consistency. There are two major truth maintenance systems: single-context and multi-context truth maintenance. In single context systems, consistency is maintained among all facts in memory (database). Multi-context systems allow consistency to be relevant to a subset of facts in memory (a context) according to the history of logical inference. This is achieved by tagging each fact or deduction with its logical history. Multi-agent truth maintenance systems perform truth maintenance across multiple memories, often located on different machines.

Many–Objective Evolutionary Optimisation

Francesco di Pierro
University of Exeter, UK

Soon-Thiam Khu
University of Exeter, UK

Dragan A. Savić
University of Exeter, UK

INTRODUCTION

Many-objective evolutionary optimisation is a recent research area that is concerned with the optimisation of problems consisting of a large number of performance criteria using evolutionary algorithms. Despite the tremendous development that multi-objective evolutionary algorithms (MOEAs) have undergone over the last decade, studies addressing problems consisting of a large number of objectives are still rare. The main reason is that these problems cause additional challenges with respect to low-dimensional ones. This chapter gives a detailed analysis of these challenges, provides a critical review of the traditional remedies and methods for the evolutionary optimisation of many-objective problems and presents the latest advances in this field.

BACKGROUND

There has been considerable recent interest in the optimisation of problems consisting of more than three performance criteria, realm that was coined many-objective optimisation by Farina and Amato (Farina, & Amato, 2002). To date, the vast majority of the literature has focused on two and three-dimensional problems (Deb, 2001). However, in recent years, the incorporation of multiple indicators into the problem formulation has clearly emerged as a prerequisite for a sound approach in many engineering applications (Coello Coello, Van Veldhuizen, & Lamont, 2002). Despite the tremendous development that MOEAs have undergone over the last decade, and their ample success in disparate applications, studies addressing high-dimensional real-life problems are still rare (Coello Coello, & Aguirre, 2002). The main reason is that

many-objective problems cause additional challenges with respect to low-dimensional ones:

If the dimensionality of the objective space increases, then in general, the dimensionality of the Pareto-optimal front also increases.

The number of points required to characterise the Pareto-optimal front increases exponentially with the number of objectives considered.

It is clear that these two features represent a hindrance for most of the population-based methods, including MOEAs. In fact, in order to provide a good approximation of a high-dimensional optimal Pareto front, this class of algorithms must evolve populations of solutions of considerable size. This has a profound impact on their performance, since evaluating each individual solution may be a time-consuming task. Using smaller populations would not be a viable option, at least for Pareto-based algorithms, given the progressive loss of selective pressure they experience as the number of objectives increases, with a consequent deterioration of performances, as it is theoretically shown in (Farina, & Amato, 2004) and empirically evidenced in (Deb, 2001, pages 404-405). In contrast to Pareto-based methods, traditional multi-objective optimisation approaches, which work by reducing the multi-objective problem into a series of parameterised single-objective ones that are solved in succession, are not affected by the curse of dimensionality. However, such strategies cause each optimisation to be executed independent to each other, thereby losing the implicit parallelism of population-based multi-objective algorithms.

The remainder of this chapter provides a detailed review of the methods proposed to address the first two

issues affecting many-objective evolutionary optimisation and discusses the latest advances in the field.

REMEDIAL MEASURES: STATE-OF-THE-ART

The possible remedies that have been proposed to address the issues arising in evolutionary many-objective optimisation can be broadly classified as follows:

- aggregation, goals and priorities
- conditions of optimality
- dimensionality reduction

In the next sub-sections we give an overview of each of these methods and review the approaches that have been so far proposed.

Aggregation, Goals and Priorities

This class of methods tries and overcome the difficulties described in the previous section through the decomposition of the original problem into a series of parameterised single-objective ones, that can then be solved by any classical or evolutionary algorithm.

Many aggregation-based methods have been presented so far and they are usually based on modifications of the weighted sum approach, such as the augmented Tchebycheff function, that are able to identify exposed solutions, and explore non-convex regions of the trade-off surface. However, the problem of selecting an effective strategy to vary weights or goals so that a representative approximation of the trade-off curve can be achieved is still unresolved.

The ε-constraint approach (Chankong, & Haimes, 1983), which is based on minimisation of one (the most preferred or primary) objective function while considering the other objectives as constraints bound by some allowable levels, was also used in the context of evolutionary computing. The main limitation of this approach is its computational cost and the lack of an effective strategy to vary bound levels (ε). Recently, Laumanns *et al.* (Laumanns, Thiele, & Zitzler, 2006) proposed a variant of the original approach where they developed a variation scheme based on the concept of ε-Pareto dominance (efficiency) (White, 1986) that adaptively generates constraint values, thus enabling the exhaustive exploration of the Pareto front, provided

the scheme is coupled with an exact single-objective optimiser. It must be pointed out however, that none of the methods described above has ever been thoroughly tested in the context of many-objective optimisation.

The Multiple Single Objective Pareto Sampling (MSOPS 1 & 2), an interesting hybridisation of the aggregation method with goal specification, was presented in (Hughes, 2003, Hughes, 2005). In the MSOPS, the selective pressure is not provided by Pareto ranking. Instead, a set of user defined target vectors is used in turn, in conjunction with an aggregation method, to evaluate the performance of each solution at every generation of a MOEA. The greater is the number of targets that a solution nears, the better its rank. The authors suggested two aggregation methods: the weighted min-max approach (implemented in MSOPS) and the Vector-Angle-Distance-Scaling (implemented in MSOPS 2). The results indicated with statistical significance that NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002), the Pareto-based MOEA used for comparative purposes, was outperformed on many objective problems. This was also recently confirmed by Wagner *et al.* in (Wagner, Beume, & Naujoks, 2007), where they benchmarked traditional MOEAs, aggregation-based methods and indicator-based methods on a up to 6-objective problems and suggested a more effective method to generate the target vectors.

Conditions of Optimality

Recently, great attention has been given to the role that conditions of optimality may play in the context of many-objective evolutionary optimisation when used to rank trial solutions during the selection stage of MOEA in alternative to, or conjunction with, Pareto efficiency. Farina *et al.* (Farina, & Amato, 2004) proposed the use of a fuzzy optimality condition, but did not provide a direct means to incorporate it into a MOEA. Köppen *et al.* (Koppen, Vincente-Garcia, & Nickolay, 2005) also suggested the fuzzification of the Pareto dominance relation, which was exploited within a generational elitist genetic algorithm on a synthetic MOP. The concept of *knee* (Deb, 2003), has also been exploited in the context of evolutionary many-objective optimisation. Simply stated, a knee is a portion of a Pareto surface where the marginal substitution rates are particularly high, i.e. a small improvement in one objective lead to a high deterioration of the others. A graphical representation is given in Figure 1. The idea

M

is that, with no prior information about the preference structures of the DM, the knee is likely to be the most interesting area. Branke *et al*. (Branke, Deb, Dierolf, & Osswald, 2004) developed two methodologies to detect solutions laying on knees, and incorporated them into the second stage (crowding measure) of the ranking procedure of NSGA-II. The first methodology consists in evaluating for each individual in a population the angle between itself and some neighbouring solutions and to use this value to favour solutions with higher angles, i.e. closer to the knee. The methodology, however, scales poorly with the number of objectives. The second strategy resorts to the expected marginal utility function to detect solutions close to the knee. This approach extends easily with the number of objectives; however, the sampling necessary to evaluate the expectation of the marginal utility function with a certain confidence may become expensive. Neither of these approaches has been tested on many-objective problems.

The concept of approximate optimal solutions has also been investigated to some extent in the context of evolutionary many-objective optimisation. In particular, ε-efficiency was considered to be potentially effective to ease some of the difficulties associated with many-objective problems. A recent study by Wagner *et al.* (Wagner, Beume, & Naujoks, 2007) showed the excellent performance of ε-MOEA (Deb, Mohan, & Mishra, 2003) on a 6-objective instance of two synthetic test functions. A good review on the application of approximate conditions of optimality is given in (Burke, & Landa Silva, 2006), where the authors also compared the effect of using two relaxed forms of Pareto dominance as evaluation methods within two MOEAs.

Recently,di Pierro *et al.* (di Pierro, Khu, & Savić, 2007) proposed a ranking scheme based on Preference Ordering (Das, 1999), a condition of optimality that generalises Pareto efficiency, but it is more stringent, and tested it using NSGA-II as the optimisation shell on a suit of seven benchmark problems with up to eight objectives. Results indicated that the methodology proposed enhanced significantly the convergence properties of the standard NSGA-II algorithm on all the test problems. The strengths of this approach are its absence of parameters to tune and the fact that it showed very good performances across varying problem features; the drawbacks its computation runtime and

Figure 1. Simple Pareto front with a knee
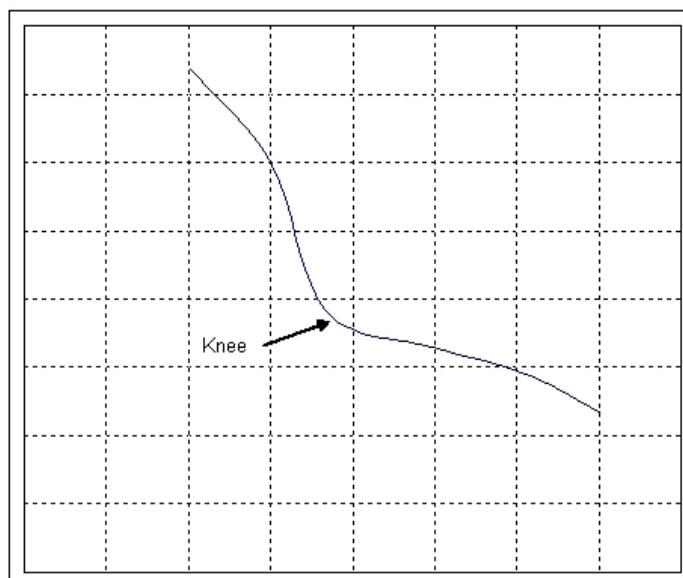
the fact that its combination with diversity preserving mechanisms that favours extreme solutions may ingenerate too high of a selective pressure.

In (Sato, Aguirre, & Tanaka, 2007), Sato *et al*. introduced an approach to modify the Pareto dominance condition used within the selection stage of Pareto-based MOEAs, by which the area dominated by any given point is contracted or expanded according to a formula derived from the Sine theorem and the extent of the contraction/expansion is controlled by a constant factor. The results of a series of experiments performed using NSGA-II equipped with the contraction/expansion mechanism on 0/1 multi-objective knapsack problems showed substantially improved convergence and diversity performances compared to the standard NSGA-II algorithm. However, it was also shown that the optimal value for the contraction/expansion factor depends strongly on various problem features, and no indication was given to support a correct choice.

Most modern MOEAs rely on a two-stage ranking during the selection process. At the first stage the ranks are assigned according to some form of Pareto-based dominance relation; if ties exist, these are resolved resorting to mechanisms that favour a good distribution along the Pareto front of the solutions examined. It has now been acknowledged that this second stage of the ranking procedure may in fact be detrimental in case of many objectives, as it was shown in (Purshouse, & Fleming, 2003b) for the case of NSGA-II. Recent efforts have therefore focused on replacing diversity preserving mechanisms at this second stage with more effective ones. Koppen and Yoshida (Koppen, & Yoshida, 2007) proposed four secondary ranking assignments and tested them by replacing the crowding distance assignment within NSGA-II. The results indicated improved convergence in all cases compared to the standard NSGA-II. However, the authors did not report any result on the diversity performance of the algorithms.

Dimensionality Reduction Methods

The aim of this class of methods is usually to transform the objective space into a lower dimension representation, either one-off (prior to the optimisation) or iteratively (as the search progresses).

Deb and Saxena (Deb, & Saxena, 2006) developed a procedure based on principal component analysis (PCA) for reducing the dimension of the problem to solve. The procedure consists in performing a series of optimisations using a state-of-the-art MOEA, each one focusing only on the objectives that PCA found explaining most of the variance on the basis of Pareto front obtained with the previous optimisation. Recently Saxena and Deb (Saxena, & Deb, 2007) extended their work and replaced PCA with two dimensionality reduction techniques, the correntropy PCA and a modified maximum variance unfolding that could also detect non-linear interactions in the objective space. The results indicated that the former method suffered to some extent from a difficult choice o the best kernel function to use, whereas for the latter, the authors performed a significant number of experiments to suggest bound values of the only free parameter of the procedure. It must be highlighted that these two studies are the only efforts that have challenged new algorithms on highly-dimensional test problems (up to 50 objectives).

In a recent study Brockhoff and Zitzler (Brockhoff, & Zitzler, 2006b) introduced the minimum objective subset problem (MOSS), which is concerned with the identification of the largest set of objectives that can be removed without altering the dominance structure of the problem (i.e. the set of Pareto optimal solutions obtained considering all the objectives or only the MOSS is the same), and developed an exact algorithm and a greedy heuristic to solve it. Subsequently (Brockhoff, & Zitzler, 2006a), they proposed a measure of variation for the dominance structure and extended the MOSS to allow for dimensionality reductions involving predefined thresholds of problem structure changes. However, they did not propose a mechanism to incorporate these algorithms within a MOEA.

Recently, the analysis of the relationships of interdependence between the objectives of an optimisation problem has been successfully exploited to devise effective reduction methods. Following the definitions of conflict, support or harmony, and independence proposed in (Carlsson, & Fuller, 1995), Purshouse and Fleming (Purshouse, & Fleming, 2003a) discussed the effects of these relationships in the context of many-objective evolutionary optimisation. In a later study Purshouse and Fleming (Purshouse, & Fleming, 2003c) also suggested, in the case of objectives independence, a divide-and-conquer algorithm based on objective space decomposition.

FUTURE TRENDS

As it appears from the discussion above, there is an increasing effort to develop strategies that are able to overcome the limitations of Pareto-based methods when solving problems with many objectives. Although promising results have been generally reported, most of the approaches presented are of an empirical nature, which makes it difficult to draw conclusions that can be generalised.

   With the exception of dimensionality reduction techniques, the majority of the studies presented to date focus on mechanisms to improve the ranking of the solutions in the selection process. However, the analysis of these mechanisms is usually undertaken in isolation with respect to the other components of the algorithms. In our view, this is an important limitation that next generation algorithms will have to address, in particular, by undertaking the analysis of these mechanisms in relation with the variation operators.

   Moreover, there has been little attention in trying to characterise the solutions that a given method (belonging to the first or second category identified in the previous section) favours, in relation to the properties of the problem being solved. Theoretical frameworks are therefore needed in order to analyse existing methods and develop more focused approaches. As it was pointed out by di Pierro in (di Pierro, 2006), where he provided a theoretical framework to analyse the effect of the Preference Ordering-based ranking procedure in relation to the interdependence relationships a problem, this approach enables predicting the effect of applying a given methodology to a particular problem with limited prior knowledge, which is certainly an advantage since the goal of developing powerful algorithms is to solve (often for the first time) real life problems.

CONCLUSIONS

In this chapter we have provided a comprehensive review of the state-of-the-art of evolutionary algorithms for the optimisation of many objective problems discussing limitations and strengths of the approaches described, and we have suggested future trends of research for a field that is gathering increasing momentum.

REFERENCES

Branke, J., Deb, K., Dierolf, H., and Osswald, M., (2004). Finding Knees in Multi-objective Optimization, Kanpur Genetic Algorithm Laboratory, KanGAL Tech. Rep. No. 2004010.

Brockhoff, D. and Zitzler, E., (2006a). Dimensionality Reduction in Multiobjective Optimization with (Partial) Dominance Structure Preservation: Generalized Minimum Objective Subset Problems, TIK-Report No. 247,.

Brockhoff, D. and Zitzler, E., (2006b). On Objective Conflicts and Objective Reduction in Multiple Criteria Optimization, TIK-Report No. 243, TIK-Report No. 243.

Burke, E. K., & Landa Silva, J. D., (2006). The Influence of the Fitness Evaluation Method on the Performance of Multiobjective Search Algorithms, *European Journal of Operational Research*, (169) 3, 875-897.

Carlsson, C., & Fuller, R., (1995). Multiple Criteria Decision Making: The Case for Interdependence, *Computers and Operations Research*, (22) 251-260.

Chankong, V., & Haimes, Y. Y., (1983). *Multiobjective decision making: theory and methodology*, Dover Publications.

Coello Coello, C. A., & Aguirre, A. H., (2002). Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization Approach, *Artificial Intelligence for Engineering Design, Analysis and Manufacture*, (16) 1, 39-53.

Coello Coello, C. A., Van Veldhuizen, D. A., & Lamont, G. B., (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers.

Das, I., (1999). A Preference Ordering Among Various Pareto Optimal Alternatives, *Structural Optimization*, (18) 1, 30-35.

Deb, K., (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T., (2002). A Fast and Elitist Multiobjective Genetic Al-

gorithm: NSGA-II, *IEEE Transaction on Evolutionary Computation*, (6) 2, 182-197.

Deb, K., & Saxena, D. K., (2006). "Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems", in IEEE Congress on Evolutionary Computation, (CEC 2006), pp. 3353-3360.

Deb, K., (2003). "Multi-objective evolutionary algorithms: Introducing bias among Pareto-optimal solutions", in *Advances in Evolutionary Computing: Theory and Applications*. A. Ghosh and S. TsuTsui, Eds. London: Springer -Verlag, pp. 263-292.

Deb, K., Mohan, M., and Mishra, S., (2003). A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pereto-Optimal Solutions, Kanpur Genetic Algorithm Laboratory, KanGAL Technical Report, 2003002.

di Pierro, F., (2006). Many-Objective Evolutionary Algorithms and Applications to Water Resources Engineering, School of Engineering, Computer Science and Mathematics, University of Exeter, UK, Ph.D. Thesis.

di Pierro, F., Khu, S.-T., & Savic, D. A., (2007). An Investigation on Preference Order - Ranking Scheme for Multi Objective Evolutionary Optimisation, *IEEE Transaction on Evolutionary Computation*, (11) 1, 17-45.

Farina, M., & Amato, P., (2004). A Fuzzy Definition of Optimality for Many-Criteria Optimization Problems, *IEEE Transactions on Systems, Man, and Cybernetics Part A---Systems and Humans*, (34) 3, 315-326.

Farina, M., & Amato, P., (2002). "On the Optimal Solution Definition for Many-criteria Optimization Problems", in Proceedings of the NAFIPS-FLINT International Conference 2002, pp. 233-238, Piscataway, New Jersey: IEEE Service Center.

Hughes, E. J., (2003). Multiple single objective pareto sampling, in 2003 Congress on Evolutionary Computation, pp. 2678-2684.

Köppen, M., Vincente-Garcia, R., & Nickolay, B., (2005). Fuzzy-Pareto-Dominance and Its Application in Evolutionary Multi-objective Optimization, in Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005, vol. 3410, pp. 399-412.

Köppen, M., & Yoshida, K., (2007). Substitute Distance Assignments in NSGA-II for Handling Many-Objective Optimization Problems, in Evolutionary Multi-Criterion Optimization, .4th International Conference, EMO 2007, S. Obayashi and K. Deb and C. Poloni and T. Hiroyasu and T. Murata Eds. Springer.

Laumanns, M., Thiele, L., & Zitzler, E., (2006). An efficient, adaptive parameter variation scheme for metaheuristic based on the epsilon-constraint method, *European Journal of Operational Research*, (169) 932-942.

Purshouse, R. C., & Fleming, P. J., (2003c). An adaptive divide-an-conquer methodology for evolutionary multi-criterion optimisation, in Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), pp. 133-147, Berlin: Springer.

Purshouse, R. C., & Fleming, P. J., (2003a). Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation, in Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), pp. 16-30, Berlin: Springer.

Purshouse, R. C., & Fleming, P. J., (2003b). Evolutionary Multi-Objective Optimisation: An Exploratory Analysis, in Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003), vol. 3, pp. 2066-2073 IEEE Press.

Sato, H., Aguirre, H. E., & Tanaka, K., (2007). Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs, in Evolutionary Multi-Criterion Optimization, .4th International Conference, EMO 2007, pp. 5-19.

Saxena, D. K., & Deb, K., (2007). Non-linear Dimensionality Reduction Procedures, in Evolutionary Multi-Criterion Optimization, .4th International Conference, EMO 2007, pp. 772-787, S. Obayashi and K. Deb and C. Poloni and T. Hiroyasu and T. Murata Eds. Springer.

Wagner, T., Beume, N., & Naujoks, B., (2007). Pareto-, Aggregation-, and Indicator-based Methods in Many-Objective optimization, in Evolutionary Multi-Criterion Optimization.4th International Conference, EMO 2007,

pp. 742-756, S. Obayashi and K. Deb and C. Poloni and T. Hiroyasu and T. Murata Eds. Springer.

White, D. J., (1986). Epsilon Efficiency, *Journal of Optimization Theory and Applications*, (49) 319-337.

KEY TERMS

Evolutionary Algorithms: Solution methods inspired by the natural evolution process that evolve a population of solutions to an optimisation problem through iterative application of randomised processes of recombination and selection, until a termination criteria is met.

Many-Objective Problem: Problem consisting of more than 3-4 objectives to be concurrently maximised/minimised.

Pareto Front: Image of the Pareto Set onto the performance (objective) space.

Pareto Optimal Solution: Solution that is not dominated by any other feasible solution.

Pareto Set: Set of Pareto Optimal solutions.

Ranking Scheme: Scheme that assigns to each solution of a population a score that is a measure of its fitness relative to the other members of the same population.

Selective Pressure: The ratio between the number of expected selections of the best solution and the mean performing one.

Mapping Ontologies by Utilising Their Semantic Structure

Yi Zhao
Fernuniversitaet in Hagen, Germany

Wolfgang A. Halang
Fernuniversitaet in Hagen, Germany

INTRODUCTION

As a key factor to enable interoperability in the Semantic Web (Berners-Lee, Hendler & Lassila, 2001), ontologies are developed by different organisations at a large scale, also in overlapping areas. Therefore, ontology mapping has come into forth to achieve knowledge sharing and semantic integration in an environment where knowledge and information are represented by different underlying ontologies.

The ontology mapping problem can be defined as acquiring the relationships that hold between the entities of two ontologies. Mapping results can be used for various purposes such as schema/ontology integration, information retrieval, query mediation, or web service mapping.

In this article, a method to map concepts and properties between ontologies is presented. First, syntactic analysis is applied based on token strings, and then semantic analysis is executed according to WordNet (Fellbaum, 1999) and tree-like graphs representing the structures of ontologies. The experimental results exemplify that our algorithm finds mappings with high precision.

BACKGROUND

Borrowed from philosophy, ontology refers to a systematic account of what can exist or 'be' in the world. In the fields of artificial intelligence and knowledge representation, ontology refers to the construction of knowledge models that specify a set of concepts, their attributes, and the relationships between them. Ontologies are defined as "explicit conceptualisation(s) of a domain" (Gruber, 1993), and are seen as a key to realise the vision of the Semantic Web.

Ontology, as an important technique to represent knowledge and information, allows to incorporate semantics into data to drastically enhance information exchange. The Semantic Web (Berners-Lee, Hendler & Lassila, 2001) is as a universal medium for data, information, and knowledge exchange. It suggests to annotate web resources with machine-processable metadata. With the rapid development of the Semantic Web, it is likely that the number of ontologies used will strongly increase over the next few years. By themselves, however, ontologies do not solve any interoperability problem. Ontology mapping (Ehrig, 2004) is, therefore, a key to exploit semantic interoperability of information and, thus, has been drawing great attention in the research community during recent years. This section introduces the basic concepts of information integration, ontologies, and ontology mapping.

Mismatches between ontologies are mainly caused by independent development of ontologies in different organisations. They become evident when trying to combine ontologies which describe partially overlapping domains. The mismatches between ontologies can broadly be distinguished into syntactic, semantic, and structural heterogeneity. Syntactic heterogeneity denotes differences in the language primitives used to specify ontologies, semantic heterogeneity denotes differences in the way domains are conceptualised and modelled, while structural heterogeneity denotes differences in information structuring.

There have been a number of previous works proposed so far on ontology mapping (Shvaiko, 2005, Noy, 2004, *Sabou, 2006, Su, 2006*). In (Madhavan, 2001), a hybrid similarity mapping algorithm has been introduced. The proposed measure integrates the linguistic and structural schema matching techniques. The matching is based primarily on schema element names, not considering their properties. LOM (Li, 2004) is a semi-automatic lexicon-based ontology-mapping tool that supports a human mapping engineer with a

first-cut comparison of ontological terms between the ontologies to be mapped. It decomposes multi-word terms into their word constituents except that it does not perform direct mapping between the words. The procedure associates the WordNet synset index numbers of the constituent words with ontological term. The two terms which have the largest number of common synsets are recorded and presented to the user.

MAIN FOCUS OF THE CHAPTER

Our current work tries to overcome the limitations mentioned above, and to improve precision of ontology mapping. The research goal is to develop a method and to evaluate results of ontology mappings.

In this article, we present a method to map ontologies synthesised of token-based syntactic analysis, and semantic analysis employing the WordNet (Fellbaum, 1999) thesaurus and tree-structured graphs. The algorithm is outlined and expressed in pseudo-code as listed in Figure 1. The promising results obtained from experiments indicate that our algorithm finds mappings with high precision.

Syntax-Level Mapping Based on Tokenisation

Before employing syntactic mapping, a pre-processing is inevitable, which is called tokenisation. Here, ontologies are represented in the language OWL-DL[1]. Therefore, all ontology terms are represented with OWL URI. For example, in ontology "beer", an OWL Class 'Ingredient' is described by

"[OWLClassImpl] http://www.purl.org/net/ontology/beer#Ingredient",

where "*[OWLClassImpl]*" implies OWL class, URL "http://www.purl.org/net/ ontology/beer" addresses the provenance of the ontology, and 'Ingredient' is the class name. Tokenisation should first extract the valid ontology entities from OWL descriptions, which, in this example, is 'Ingredient'.

Moreover, the labels of ontology entities (classes

Figure 1. Pseudo-code of mapping algorithm

```
Input: OWL O1, OWL O2, threshold sigma;
Output: similarity between O1 and O2;
Begin
build tree-structured graphs for O1 and O2, and get their edge sets E1 and E2;
for each child node Ci ∈ E1 do
    for each child node Cj ∈ E2 do
        tokenise Ci and Cj into token sets tci and tcj;
        if (tci unequal to tcj) then

            calculate syntactic level similarity Sim_syn between tci and tcj;

            if ( Sim_syn < sigma) then // semantic mapping
                compute semantic-level similarity of tci, tcj based on WordNet;
                if (tci and tcj have no WordNet relationship) then

                    determine similarity Sim_tsg with the specific properties and

                    relationships between their parent/child nodes in ts-graphs
                fi
            fi
        fi
    od
od
end
```

and properties) are quite often defined with different representations by different organisations. For instance, representations may be with or without connector symbols, with upper or lower cases, etc., which renders it very complicated and hard to identify terms. Tokenisation means to parse names into tokens based on specific rules or symbols by customisable tokenisers using punctuation, upper case, special symbols, digits, etc. In this way, a class or property name can be tokenised into one or several token strings. For example, the term 'Social_%26_Science' can be tokenised as 'Social', '26', and 'Science'. Note that the terms can sometimes contain digits like date which is not neglectable.

For simplicity, we assume that all terms of ontology concepts (classes, properties) are described without abbreviations. The mapping process between different class and property names is then transformed to mapping between tokens.

We first check whether the original child nodes are equal ignoring case. Otherwise, the tokens are used instead to check whether they are equal. If not, the similarity measure based on the edit distance is adopted to calculate similarity. If the calculated similarity value is above a threshold σ (for example, 0.95), the compared nodes are considered to be similar. The process continues to deal with the next pair of nodes in the same way.

The edit distance formulated by Levenshtein (Levenshtein, 1966) and the string mapping method proposed by Maedche & Staab (Maedche & Staab, 2002) are employed here to calculate token similarity. The edit distance is a well-established method to weigh the difference between two strings. It measures the minimum number of token insertions, deletions, and substitutions required to transform one string into another using a dynamic programming algorithm. The string matching method is used to calculate token similarity based on Levenshtein's edit distance:

$$Sim(X,Y) = \max(0, \frac{\min(|X|,|Y|) - ed(X,Y)}{\min(|X|,|Y|)}) \in [0,1]$$

(1)

where X, Y are token strings, '$|X|$' is the length of X, '$min( )$' and '$max()$' denotes the minimum/maximum value of two arguments, respectively, and '$ed( )$' is the edit distance.

As the original ontology terms may have been to-

kenised into many sub-terms, i.e., tokens, it is necessary to separately calculate similarity between each pair of token strings. Assume that the number of tokens of the first term is m, n for the second term, and assume $m \geq n$, the total similarity measure according to Eq. (1) is:

$$Sim_{syn} = \varpi_1 Sim_{orig} + \sum_{i=1}^{m} \sum_{j=1}^{n} \varpi_{ij} Sim_{ij},$$

$$\varpi_1 + \sum_{i=1}^{m} \sum_{j=1}^{n} \varpi_{ij} = 1$$

(2)

where Sim_{orig} is the similarity between the original strings, Sim_{ij} is the similarity between tokens ith and jth from two source terms, and ω_1, ω_{ij} are the weights for Sim_{orig} and Sim_{ij}. The sum of ω_{ij} and ω_1 are supposed to be 1. Given a predefined similarity threshold, if the acquired similarity value is greater than or equal to the threshold, then two tokens are considered similar, vice versa.

Semantic-Level Mapping Based on Ontology Structure

Semantic heterogeneity occurs when there is a disagreement about meaning, interpretation, or intended use of the same or related data. Semantic relations (Gahleitner, & Woess, 2004) are:

- different naming of the same content, i.e., synonyms,
- different abstraction levels: generic terms vs. more specific ones (name vs. first name and last name), hypernyms or hyponyms, and
- different structures about the same content (separate type vs. part of a type), i.e., meronyms.

In ontology mapping, WordNet is one of the most frequently used sources of background knowledge. Actually, it plays the role of an 'intermediate' to help finding semantic heterogeneity. The WordNet library can be accessed with the Java API JWNL[2]. It groups English words into sets of synonyms called synsets providing short, general definitions, and it records the various semantic relations between these synonym sets. The purpose is to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications. It is assumed that each sense in a

WordNet synset describes a concept. WordNet senses are related among themselves via synonym, hyponym, and hyperonym relations. Terms lexicalising the same concept (sense) are considered to be equivalent through the synonym relationship, while hypernyms, hyponyms, and meronyms are considered similar. With this rule, the original ontology terms and their relative token strings are first checked whether they have the same part of speech, i.e., noun, verb, adjective, adverb. The next step is to judge whether they are synonyms, hypernyms, hyponyms, or meronyms. Analogue to Eq. (2), for a pair of nodes a similarity value is calculated according to the weights of different similarity parts. If it exceeds a given threshold, the pair is considered to be similar.

If the above-mentioned syntactic and semantic WordNet mapping methods still could not find a mapping between two terms, another semantic-level method based on tree-structured graphs is applied.

As rendered with SWOOP[3] (a hypermedia-inspired ontology browser and editor based on OWL ontologies, which supports renderers to obtain class/property hierarchy trees as well as definitions of and inferred facts on OWL classes and properties), ontology entities are represented by class/property hierarchy trees. From class hierarchy trees, tree-structured graphs are constructed. Based on the notion of structure graphs (Lian, 2004), a tree-structured graph (*ts*-graph) is defined as:

Definition 1. Given a tree of sets T, N the union of all sets in T, and E the set of edges in T, then *ts-g* $(T) = (N, E)$ is called tree-structured graph of T, if it holds $(a, b) \in E$ if and only if a is a parent element of b; a is called parent node, and b child node.

In building a *ts*-graph, breadth-first traversal is applied to traverse a tree hierarchy. To construct a *ts*-graph, we begin with the root node's first child node. All its child nodes and their relative parent nodes form edges as (parent node, child node) for the *ts*-graph. The process is repeated until the tree is completely traversed.

After the *ts*-graphs of two ontologies to be matched are built, the edge sets of both graphs are employed in the mapping process. The relative positions of a pair's (from two ontologies) nodes within their tree-structured graphs determines the semantic-level mapping between them. In Table 1 we summarise three types of relationships between edges characterising properties, child classes, and parent-and-child classes. By understanding these properties, we can derive that entities having the same properties are similar. This is not a rule always holding true, but it is a strong indictor for similarity.

Experimental Results

The implementation of our algorithm was written in Java. It relies on SWOOP for parsing OWL files. All tests were run on a standard PC (with Windows XP). Our test cases include ontologies "Baseball team", and "Russia" provided by the institute AIFB[4].

To get an impression of the matchmaker's performance different measures have to be considered. There are various ways to measure how well retrieved information matches intended information. To evaluate

Table 1. Relationships between classes

No.	Characteristic	Rules: Given two classes *a* and *b*
R1	Properties	If properties (data type property/object property $\notin$ null) of *a* and *b* are similar, *a* and *b* are also similar.
R2	Child classes	If all child classes of *a* and *b* are similar, *a* and *b* are also similar.
R3	Parent-and-child classes	If parent class and one of the child classes of *a* and *b* are similar, *a* and *b* are also similar.

Figure 2. Analysis of mapping results with ontologies Russia
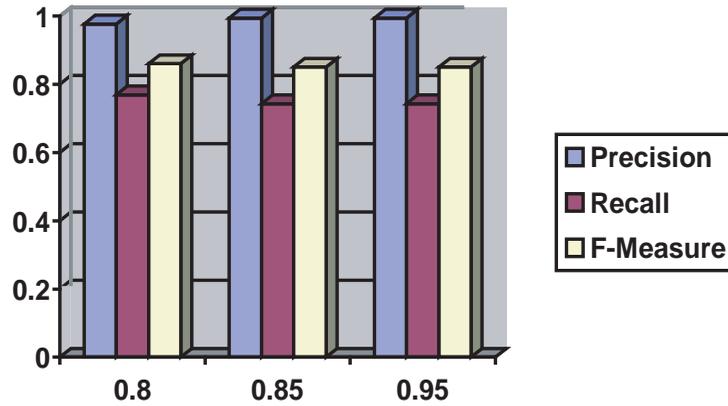
the quality of mapping results, we use standard information-retrieval metrics: *Recall* (r), *Precision* (p), and *F-Measure* (Melnik & Rahm, 2002), where *Recall* is the ratio of the number of relevant entities retrieved to the total number of relevant entities, *Precision* is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved, and

$$F\text{-}Measure = \frac{2rp}{r + p}$$

(3)

As shown in Figure 2, with the increase of the threshold, the mapping precision for "Russia" gets higher. The shortcoming of this method is its efficiency. Though we trade off efficiency to get more effective mapping results, our algorithm is still applicable to some offline web applications, like information filtering (Hanani, 2001) according to users' profile.

FUTURE TRENDS

Currently, the results of similarity computations are provided in form of text documents. In order to present mapping results more reasonably and understandably, the objective of our future work is to treat the results of similarity computations as ontology. Another objective

of our future work is to address security problems connected to ontology mapping, such as trust management in the application of web services.

CONCLUSION

The overall research goal presented in this article is to develop a method for ontology mapping that combines syntactic analysis measuring the difference between tokens by the edit distance with semantic analysis based on WordNet as semantic relation and the similarity of structured graphs representing the ontologies being compared. Empirically we have shown that our synthesised mapping method works with relatively high precision.

REFERENCES

Berners-Lee, T., Hendler J., & Lassila O. (2001). The Semantic Web. *Scientific American*, 284(5), 35-43.

Do, H., Melnik, S., & Rahm, E. (2002). Comparison of Schema Matching Evaluations. In *Proceedings of the 2nd International Workshop on Web Databases* (*German Informatics Society*).

Ehrig, M., Sure, Y.: Ontology Mapping - An Integrated

Approach. In Bussler, C., Davis, J., Fensel, D., Studer, R., eds.: *Proceedings of the 1st ESWS*. Volume 3053 of Lecture Notes in Computer Science., Heraklion, Greece, Springer Verlag, pp. 76–91, 2004.

Fellbaum, C. (1999). Wordnet: An Electronic Lexical Database. MIT press.

Gahleitner, E., & Woess, W. (2004). Enabling Distribution and Reuse of Ontology Mapping Information for Semantically Enriched Communication Services. In *Proceedings of the 15th International Workshop on Database and Expert Systems Applications* (DEXA'04).

Gruber, T.R. (1993). A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition*, 5(2), 199-220.

Hanani, U., Shapira, B., Shoval, P. (2001) Information Filtering: Overview of Issues, Research and Systems. *User Modeling and User-Adapted Interaction*, 11, 203-259.

Kalyanpur, A., Parsia, B., & Hendler, J. (2005). A Tool for Working with Web Ontologies. *Intrnational Journal on Semantic Web and Information Systems,* 1(1).

Levenshtein, I.V. (1966). Binary Codes Capable of Correcting Celetions, Insertions, and Reversals. *Cybernetics and Control Theory*, 10(8), 707-710.

Li, J. (2004). LOM: A Lexicon-based Ontology Mapping tool. In *Proceeding of the Performance Metrics for Intelligent Systems* (*PerMIS'04*), *Information Interpretation and Integration Conference* (*I³CON*), Gaithersburg, MD. Retrieved Aug. 30, 2007, from http://reliant.teknowledge.com/DAML/I3con.pdf.

Lian, W., Cheung, D.W., & Yiu, S.M. (2004). An Efficient and Scalable Algorithm for Clustering XML Documents by Structure. *IEEE Transactions on Knowledge and Data Engineering*, 16(1), 82-96.

Maedche A., & Staab, S. (2002) Measuring Similarity between Ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management* (EKAW-2002). LNCS/LNAI 2473, pp. 251-263, Springer-Verlag.

Noy, N.F. (2004). Semantic Integration: A Survey of Ontology-based Approaches. *SIGMOD Record*, 33(4), 65-70.

Sabou, M., d'Aquin, M., & Motta, E. (2006) Using the Semantic Web as Background Knowledge for Ontology Mapping. In *Proceedinds of the International Workshop on Ontology Matching* (OM-2006), collocated with ISWC'06.

Su, X.M. & Atle Gulla, J. (2006) An Information Retrieval Approach to Ontology Mapping. *Data & Knowledge Engineering* 58(1), 47-69.

KEY TERMS

Ontology: As a means to conceptualise and structure knowledge, ontologies are seen as the key to realise the vision of the semantic web

Ontology Mapping: Ontology mapping is required to achieve knowledge sharing and semantic integration in an environment with different underlying ontologies.

Precision: The ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved.

Recall: The ratio of the number of relevant entities retrieved to the total number of relevant entities.

Semantic Web: Envisioned by Tim Berners-Lee, the semantic web is as a universal medium for data, information, and knowledge exchange. It suggests to annotate web resources with machine-processable metadata.

Similarity Measure: A method used to calculate the degree of similarity between mapping sources.

Tokenisation: Tokenisation extracts the valid ontology entities from OWL descriptions.

Tree-Structured Graph: A graphical structure to represent a tree with nodes and a hierarchy of its edges.

ENDNOTE

1.   http://www.w3.org/TR/owl-features/
2.   http://sourceforge.net/projects/jwordnet
3.   www.mindswap.org/2004/SWOOP/

4 The datasets are available from http://www.aifb. uni-karlsruhe.de/WBS/meh/mapping/.

Mathematical Modeling of Artificial Neural Networks

Radu Mutihac
University of Bucharest, Romania

INTRODUCTION

Models and algorithms have been designed to mimic information processing and knowledge acquisition of the human brain generically called artificial or formal neural networks (ANNs), parallel distributed processing (PDP), neuromorphic or connectionist models. The term network is common today: computer networks exist, communications are referred to as networking, corporations and markets are structured in networks. The concept of ANN was initially coined as a hopeful vision of anticipating artificial intelligence (AI) synthesis by emulating the biological brain.

ANNs are alternative means to symbol programming aiming to implement neural-inspired concepts in AI environments (neural computing) (Hertz, Krogh, & Palmer, 1991), whereas cognitive systems attempt to mimic the actual biological nervous systems (computational neuroscience). All conceivable neuromorphic models lie in between and supposed to be a simplified but meaningful representation of some reality. In order to establish a unifying theory of neural computing and computational neuroscience, mathematical theories should be developed along with specific methods of analysis (Amari, 1989) (Amit, 1990). The following outlines a tentatively mathematical-closed framework in neural modeling.

BACKGROUND

ANNs may be regarded as dynamic systems (discrete or continuous), whose states are the activity patterns, and whose controls are the synaptic weights, which control the flux of information between the processing units (adaptive systems controlled by synaptic matrices). ANNs are parallel in the sense that most neurons process data at the same time. This process can be synchronous, if the processing time of an input neuron is the same for all units of the net, and

asynchronous otherwise. Synchronous models may be regarded as discrete models. As biological neurons are asynchronous, they require a continuous time treatment by differential equations.

Alternatively, ANNs can recognize the state of environment and act on the environment to adapt to given viability constraints (cognitive systems controlled by conceptual controls). Knowledge is stored in conceptual controls rather than encoded in synaptic matrices, whereas learning rules describe the dynamics of conceptual controls in terms of state evolution in adapting to viability constraints.

The concept of paradigm referring to ANNs typically comprises a description of the form and functions of the processing unit (neuron, node), a network topology that describes the pattern of weighted interconnections among the units, and a learning rule to establish the values of the weights (Domany, 1988). Although paradigms differ in details, they still have a common subset of selected attributes (Jansson, 1991) like simple processing units, high connectivity, parallel processing, nonlinear transfer function, feedback paths, non-algorithmic data processing, self-organization, adaptation (learning) and fault tolerance. Some extra features might be: generalization, useful outputs from fuzzy inputs, energy saving, and potential overall high speed operation.

The digital paradigm dominating computer science assumes that information must be digitized to avoid noise interference and signal degradation. In contrast, a neuron is highly analog in the sense that its computations are based on spatiotemporal integrative processes of smoothly varying ion currents at the trigger zone rather than on bits. Yet neural systems are highly efficient and reliable information processors.

Memory and Learning

The specificity of neural processes consists in their distributive and collective nature. The phenomenon

by biological neural networks (NNs) are changing in response to extrinsic stimuli is called self-organization. The flexible nature of the human brain, represented by self-organization, seems to be responsible for the learning function which is specific to living organisms. Essentially, learning is an adaptive self-organizing process. From the training assistance point of view, there are supervised and unsupervised neural classifiers. Supervised classifiers seek to characterize predefined classes by defining measures that maximize in-class similarity and out-class dissimilarity. Supervision may be conducted either by direct comparison of output with the desired target and estimating error, or by specifying whether the output is correct or not (reinforcement learning). The measure of success in both cases is given by the ability to recover the original classes for similar but not identical input data. Unsupervised classifiers seek similarity measures without any predefined classes performing cluster analysis or vector quantization. Neural classifiers organize themselves according to their initial state, types and frequency of the presented patterns, and correlations in the input patterns by setting up some criteria for classification (Fukushima, 1975) reflecting causal mechanisms. There is no general agreement on the measure of their success since likelihood optimization always tends to favor single instance classes.

Classification as performed by ANNs has essentially a dual interpretation reflected by machine learning too. It could mean either the assignment of input patterns to predefined classes, or the construction of new classes from a previously undifferentiated instance set (Stutz & Cheesman, 1994). However, the assignment of instances to predefined classes can produce either the class that best represents the input pattern as in the classical decision theory, or the classifier can be used as a content-addressable or associative memory, where the class representative is desired and the input pattern is used to determine which exemplar to produce. While the first task assumes that inputs were corrupted by some processes, the second one deals with incomplete input patterns when retrieval of full information is the goal. Most neural classifiers do not require simultaneous availability of all training data and frequently yield error rates comparable to Bayesian methods without needing prior information. An efficient memory might store and retrieve many patterns, so its dynamics must allow for as many states of activity which are stable against small perturbations as possible. Several ap-

proaches dealing with uncertainty such as fuzzy logic, probabilistic, hyperplane, kernel, and exemplar-based classifiers can be incorporated into ANN classifiers in applications where only few data are available (Ng & Lippmann, 1991).

The capacity of analog neural systems to operate in unpredictable environments depends on their ability to represent information in context. The context of a signal may be some complex collections of neural patterns, including those that constitute learning. The interplay of context and adaptation is a fundamental principle of the neural paradigm. As only variations and differences convey information, permanent change is a necessity for neural systems rather than a source of difficulty as it is for digital systems.

MATHEMATICAL FRAMEWORK OF NEURONS AND ANNS MODELING

An approach to investigate neural systems in a general frame is the mean field theory (Cooper & Scofield, 1988) from statistical physics suited for highly interconnected systems as cortical regions are. However, there is a big gap between the formal model level of description in associative memory levels and the complexity of neural dynamics in biological nets. Neural modeling need no information concerning correlations of input data, rather nonlinear processing units and a sufficiently large number of variable parameters ensure the flexibility to adapt to any relationship between input and output data. Models can be altered externally, by adopting a different axiomatic structure, and internally, by revealing new inside structural or functional relationships. Ranking several neuromorphic models is ultimately carried out based on some measure of performance.

Neuron Modeling

Central problems in any artificial system designed to mimic NNs arise from (*i*) biological features to be preserved, (*ii*) connectivity matrix of the processing units, whose size increases with the square of their number, and (*iii*) processing time, which has to be independent of the network size. Biologically realistic models of neurons might minimally include:

- Continuous-valued transfer functions (graded response), as many neurons respond to their

input in a continuous way, though the nonlinear relationship between the input and the output of cells is a universal feature;

- Nonlinear summation of the inputs and significant logical processing performed along the dendritic tree;
- Sequences of pulses as output, rather than a simple output level. A single state variable y_j representing the firing rate, even if continuous, ignores much information (e.g., pulse phase) that might be encoded in pulse sequences. However, there is no relevant evidence that phase plays a significant role in most neuronal circuits;
- Asynchronous updating and variable delay of data processing, that is, the time unit elapsing per processing step, $t \rightarrow t + 1$, is variable among neurons;
- Variability of synaptic strengths caused by the amount of transmitter substance released at a chemical synapse, which may vary unpredictably. This effect is partially modeled by stochastic generalization of the binary neural models dynamics.

Most neuromimetic models are based on the McCulloch and Pitts (1943) neuron as a binary threshold unit:

$$ y_j (t+1) = \Theta \left(\sum_{i=1}^{N} w_i^j x_i (t) - \theta_j \right) $$

(1)

where y_j represents the state of neuron j (either 1 or 0) in response to input signals $\{x_i\}_i$, θ_j stands for a certain threshold characteristic of each neuron j, time t is considered discrete, with one time unit elapsing per processing step, and Θ is the unit step (Heaviside) function:

$$ \Theta(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} $$

(2)

The weights, w_i^j, $1 \leq i \leq N$, represent the strengths of the synapses connecting neuron i to neuron j, and may be positive (excitatory) or negative (inhibitory). The weighted sum

Figure 1. Typical sigmoid transfer functions
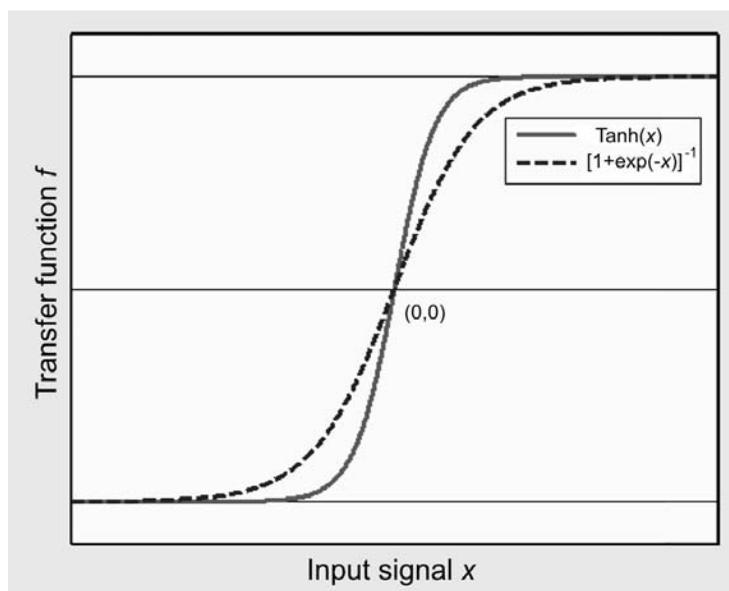
$$\sum_{i=1}^{N} w_i^j x_i (t)$$

of the inputs presented at time t to unit j must reach or exceed the threshold θ_j for the neuron j to fire. Though extremely simplified, a synchronous assembly of such formal neurons is theoretically capable of universal computation (Welstead, 1994) for suitable chosen weights $\left\{ w_i^j \right\}_{i,j}$, i.e., it may perform computations that conventional computers do, yet not necessarily so rapidly or conveniently.

A general expression that includes some of the above features derived from the digital model (1) is:

$$y_j = f_j \left(\sum_{i=1}^{N} w_i^j x_i - \theta_j \right) \tag{3}$$

where y_j is the continuous-valued state (activation) of unit j and f_j is a general transfer function. Threshold nodes are required for universal approximation and the activation function ought to be nonlinear with bounded output (Fig. 1). The neurons are updated asynchronously in random order at random times.

Mathematical Methods for ANNs Modeling

Several neural models and parallel information processing systems inspired by brain mechanisms were proposed. Almost all practical applications were achieved by simulation on conventional digital computers (von Neumann), so the real parallel processing advantages and massive unit densities hoped for were lost. Mathematical methods approaching various types of ANNs in a unified way and results from linear and nonlinear control systems used to obtain learning algorithms could be grouped in four categories:

1. Tensor products and pseudo-inverses of linear operators, which represent the specific structural connectionism and provide a mathematical explanation of the Hebbian nature of many learning algorithms (Hebb, 1949). This is due to the fact that derivatives of a wide class of nonlinear maps defined on spaces of synaptic matrices are tensor products and because the pseudo-inverse of a tensor product of linear operators is the tensor product of their pseudo-inverses;

2. Convex and nonsmooth analysis is particularly suited to nonlinear networks in proving the convergence of two main types of learning rules. The first class consists of algorithms derived from gradient methods and includes the backpropagation update rule, whereas the second class deals with algorithms based on Newton's method;

3. Control and viability theory (Aubin, 1991), which deals with neural systems that learn viable solutions as control systems satisfying given viability (state) constraints. The purpose is to derive algorithms of control systems emulated by ANNs with feedback regulation. Three classes of learning rules are envisaged: (*i*) external learning rules based on gradient methods of optimization problems involving nonsmooth functions, (*ii*) internal learning rules based on the viability theory, and (*iii*) uniform algorithms;

4. Probability theory and Bayesian statistics. Bayesian statistics and neural modeling may seem extremes of the data-modeling spectrum. ANNs are nonlinear parallel computational devices and their training by example to solve prediction and classification problems is quite a purpose-specific procedure. Contrarily, Bayesian statistics is heavily based on coherent inference and clearly defined axioms. Yet both approaches aim to create models in good accordance with the data. ANNs can be interpreted as more flexible versions of traditional regression techniques in the sense of capturing regularities in the data that the linear models are not able to handle. However, over-flexible ANNs may discover non-existent correlations in the data. Bayesian inference provides means to infer how flexible a model is warranted by the data and suppresses the tendency to assess spurious structure in the data by incorporating the Occam's razor that sets the preference for simpler models if they compete to come out with the same result. Learning in ANNs is interpreted as inference on the most probable parameters for a model, given the training data. The search in the model space can also be treated as an inference problem of relative probability for alternative models, given the data. Bayesian inference for ANNs can be implemented numerically by deterministic methods involving Gaussian approximations (MacKay, 1992), or by Monte Carlo methods (Neal, 1996).

M

Let N formal neurons link, directly for one-layer networks and indirectly for multi-layered ones, an input space X of signals to an output space Y. The state space of the system is the product $X \times Y$ of the input-output pairs (x,y), which are generically called patterns or configurations in pattern recognition (PR), data analysis, and classification problems. When $X = Y$ and the input of the patterns coincide with the outputs, (x,y), $x = y$, the system is called autoassociative; if the input and output patterns are different, (x,y) $y \neq x$, then the system is heteroassociative. Among all possible input-output patterns, a subset $K \subset X \times Y$ is chosen as training set. Most often, the input and output spaces are finite dimensional linear spaces: $X = \mathbb{R}^N$ and $Y = \mathbb{R}^M$, whereas the input signals may obey some state constraints:

- Real numbers for fuzzy applications, preferably in the intervals [0,1] or [−1, +1];
- Binary numbers that belong to {0, 1};
- Bipolar numbers that belong to {−1, +1}.

If neurons are labeled by $j = 1,2,...,N$, then let $P(N)$ of cardinal 2^N denote the family of subsets of neurons called conjuncts (or coalitions) of neurons. Any con-

nection links a postsynaptic neuron j to conjuncts $S \subset P(N)$ of presynaptic neurons. Each conjunct S preprocesses (or gates) the afferent signals $\{x_i\}_i$ produced by the presynaptic neurons through a function:

$$\{x_i\}_{i=1,2,...,N} \xrightarrow{\varphi_S} \varphi_S(x) \tag{4}$$

If conjuncts are reduced to individual neurons $S = \{i\}$, then the role of control is played by the synaptic matrix:

$$\mathbf{W} = \left\| w_j^S \right\|_{\substack{S \subset P(N) \\ j=1,2,...,N}} \tag{5}$$

where w_j^S represents the entries from S to neuron j. The modulus of the synaptic weight w_j^S represents the strength and it gives the nature of the connection from conjunct S to the formal neuron j, counted positively if the synapse is excitatory, and negatively if it is inhibitory. Accordingly, the neuron j receives the signal

$$\sum_{S \subset P(N)} w_j^S \varphi_S(x)$$

Figure 2. Conjuncts of neurons {2,3} and {4,5,6} gate the inputs to neuron 1
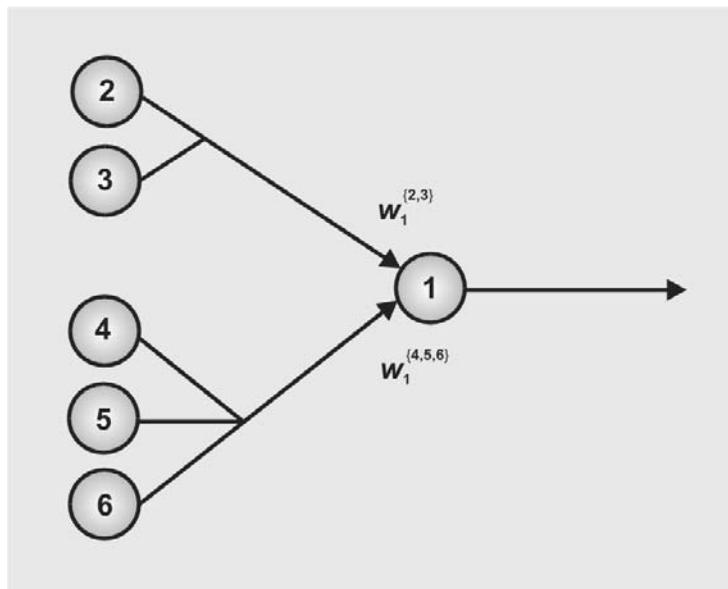
(Fig. 2), which determines its state of activity. Hence the propagation rule that characterize the network dynamics is:

$$y_j = f_j\left(\left\{w_j^S, \varphi_S(x)\right\}_{S \subset P(N)}\right)$$

(6)

for synchronous neurons (discrete dynamical system), and:

$$x_j'(t) = f_j\left(\left\{w_j^S(t), \varphi_S(x,t)\right\}_{S \subset P(N)}\right)$$

(7)

for asynchronous neurons (continuous dynamical system), where in most cases:

$$f_j\left(\left\{w_j^S, \varphi_S(x)\right\}_{S \subset P(N)}\right) = g_j\left(\sum_{S \subset P(N)} w_j^S \varphi_S(x)\right)$$

(8)

Here g_j integrates the afferent signals $\left\{w_j^S \varphi_S(x)\right\}$ sent to the neuron j by the afferent neurons through their outputs $\{x_i\}_i$, preprocessed by the conjunct S, and delivered to neuron j via the weight w_j^S. Usually, the synaptic weights $w_j^S = 0$ when $j \in S$, whereas $w_j^S \neq 0$ when $j \in S$ is associated with autoexcitation. However, when $S = \{j\}$, $S \neq \{i\}$, and $w_j^S = 0$, then the loss term $g_j\left(0, 0, ..., w_j^i \varphi_j(x), 0, ..., 0\right)$ represents some kind of forgetting like decaying frequency while the neuron in question j is not excited by the others.

Several neural systems can be expressed within this framework (Aubin, 1991).

1. Associative memories are defined by the lack of preprocessing, that is,

$$\varphi_S(x) = 0 \text{ if } |S| > 1 \text{ or } \varphi_S(x) = x_i \text{ if } S = \{i\},$$

then:

$$y_j = \sum_{i=1}^{N} w_j^i x_i + c_j$$

(9)

where $|S|$ stands for the number of elements in conjunct S.

2. Associative memories with gates are defined by preprocessing and g_j affine:

$$y_j = \sum_{i=1}^{N} w_j^S x_i + c_j$$

(10)

Boolean associative memories correspond to

$$X = \mathbb{R}^N, \quad Y = \mathbb{R}, \quad K = \{0,1\}^N \times \{0,1\},$$

and fuzzy associative memories to

$$X = \mathbb{R}^N, \quad Y = \mathbb{R}, \quad K = [0,1]^N \times [0,1],$$

hence:

$$y = \sum_{S \subset P(N)} w^S \prod_{i \in S} x_i^{\frac{1}{|S|}}$$

(11)

3. Nonlinear automata are defined by various forms of g_j:

$$f_j(x, W) = g_j\left(\sum_{S \subset P(N)} w_j^S \varphi_S(x)\right)$$

(12)

When appropriate, thresholds $\theta \in Y$ may be integrated in the processing function g:

$$g(z) = h(z - \theta)$$

(13)

If the threshold is part of the controls to be adjusted during training, it may be incorporated as an entry of an extended synaptic matrix:

$$\hat{\mathbf{W}} = \left\|\hat{w}_{ij}\right\|_{\substack{i=1,2,...,N \\ j=0,1,...,M}} \in L(\mathbb{R} \times X, Y),$$

$$\hat{w}_{ij} = \begin{cases} w_{ij} & \text{for } i = 1, 2, ..., N \text{ ; } j = 1, 2, ..., M \\ \theta_i & \text{for } i = 1, 2, ..., N \text{ ; } j = 0 \end{cases}$$

(14)

Particularly simple is the perceptron:

$$X = \mathbb{R}^N, \quad Y = \mathbb{R}, \quad \theta \in Y,$$

$$\varphi_S(x) = 0$$

if $|S| > 1$, then:

$$y = \begin{cases} 0 & \text{if } \sum_{i=1}^{N} w^i x_i < \theta \\ 1 & \text{if } \sum_{i=1}^{N} w^i x_i \geq \theta \end{cases}$$

(15)

FUTURE TRENDS

Some promising neural-inspired approaches to feature extraction and clustering were also proposed (Mao & Jain, 1995), which are adaptive online and may exhibit additional desirable properties such as robustness against outliers (Xu & Yuille, 1995) as compared to more traditional feature extraction methods.

The connection between Bayesian inference and neural models gives new perspectives to the assumptions and approximations made on ANNs and algorithms when used as associative memories. Advances in neural modeling and training algorithm design addressed dynamic range and sensitivity problems encountered by large analog systems, along with fast evolution in VLSI implementation techniques, could lead to practical real-time systems derived from the topology and parallel distributed processing performed by biological NNs.

CONCLUSION

Though ANNs are able to perform a large variety of tasks, the problems handled practically could be loosely divided into four basic types (Zupan & Gasteiger, 1994): auto- or hetero-association, classification, mapping (Kohonen, 1982), and modeling.

In neural classifiers, the set of examples used for training should necessarily come from the same (possibly unknown) distribution as the set used for testing the networks, in order to provide reliable generalization in classifying unknown patterns. Valid results are only produced if the training examples are adequately selected and their number is comparable to the number of effective parameters in the net. Quite a few classes of learning algorithms have the convergence guaranteed; moreover, they require substantial computational resources.

Generally, ANNs are used as parts of larger systems employed as preprocessing or labeling/interpretation subsystems. In many cases, the flexibility and non-algorithmic processing of ANNs surpass their incon-

veniences and make them suitable for modeling rather complex systems involving plenty of information.

REFERENCES

Amari, S. (1989). Dynamical stability of formation of cortical maps in dynamic interactions in neural networks. In Arbib M. A., & Amari, S. (Eds.) *Research work in neural computing*, *1* (pp. 15-34). Springer-Verlag.

Amit, D. J. (1990). Attractor neural networks and biological reality: Associative memory and learning. *Future Generation Computer Systems*, *6*(2), 111-119.

Aubin, J.-P. (1991). *Viability Theory*. Birkhäuser, 1991.

Cooper, L. N. & Scofield, C. L. (1988). Mean-field theory of a neural network. *Proceedings of the National Academy of Sciences of the* USA, 85(6), 1973-1977.

Domany, E. (1988). Neural networks: A biased overview. *Journal of Statistical Physics*, *51*(5-6), 743-775.

Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, *20*, 121-136.

Hebb, D. O. (1949). *The organization of behavior, A neurophysiological theory*. John Wiley & Sons Inc.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Addison-Wesley Publishing Company.

Jansson, P. A. (1991). Neural networks: An overview. *Analytical Chemistry*, *63*(6), 357-362.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59-69.

MacKay, D. J. K. (1992). A practical Bayesian framework for backpropagation networks, *Neural Computation*, *4*, 448-472.

Mao, J. & Jain, A. K. (1995). Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions of Neural Networks*, *6*(2), 296-317.

McCulloch, W. S. & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115-133.

Neal, R. M. (1996). *Bayesian learning for neural networks*. Springer-Verlag.

Ng, K. & Lippmann, R. (1991). Practical characteristics of neural network and conventional pattern classifiers. In R. Lippmann, & D. Touretzky (Eds.), *Advances in neural information processing systems* (pp. 970-976). San Francisco, CA: Morgan Kauffman Publishers Inc.

Stutz, J. & Cheesman, P. (1994). Autoclass - A Bayesian approach to classification. In J. Skilling, & S. Sibisi (Eds.), *Maximum entropy and Bayesian methods* (pp. 117-126). Cambridge: Kluwer Academic Publishers.

Sutton, R. S. & Barto, A. G. (1998). *Introduction to reinforcement learning*. Cambridge, MA: MIT Press.

Xu, L. & Yuille, A. L. (1995). Robust principal element analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, *6*(1), 131-143.

Welstead, S. T. (1994). *Neural network and fuzzy logic applications in C/C++*. John Wiley & Sons.

Zupan, J. & Gasteiger, J. (1994). Neural networks for chemists. An introduction. VCH Verlagsgesellschaft mbH, Germany.

KEY TERMS

Artificial Neural Networks (ANNs): Highly parallel networks of interconnected simple computational elements (cells, nodes, neurons, units), which mimic biological neural network.

Complex Systems: Systems made of several interconnected simple parts which altogether exhibit a high degree of complexity from each emerges a higher order behaviour.

Emergence: Modalities in which complex systems like ANNs and patterns come out of a multiplicity of relatively simple interactions.

Learning (Training) Rule: Iterative process of updating the weights from cases (instances) repeatedly presented as input. Learning (adaptation) is essential in PR where the training data set is limited and new environments are continuously encountered.

Paradigm of ANNs: Set of (i) pre-processing units' form and functions, (ii) network topology that describes the number of layers, the number of nodes per layer, and the pattern of weighted interconnections among the nodes, and (iii) learning (training) rule that specifies the way weights should be adapted during use in order to improve network performance.

Relaxation: Process by which ANNs minimize an objective function using semi- or non-parametric methods to iteratively updating the weights.

Robustness: Property of ANNs to accomplish reliable its tasks when handling incomplete and/or corrupted data. Moreover, the results should be consistent even if some part of the network is damaged.

Self-Organization Principle: A process in which the internal organization of a system that continually interacts with its environment increases in complexity without being driven by an outside source. Self-organizing systems typically exhibit emergent properties.

M

Microarray Information and Data Integration Using SAMIDI

Juan M. Gómez
Universidad Carlos III de Madrid, Spain

Ricardo Colomo
Universidad Carlos III de Madrid, Spain

Marcos Ruano
Universidad Carlos III de Madrid, Spain

Ángel García
Universidad Carlos III de Madrid, Spain

INTRODUCTION

Technological advances in high-throughput techniques and efficient data gathering methods, coupled computational biology efforts, have resulted in a vast amount of life science data often available in distributed and heterogeneous repositories. These repositories contain information such as sequence and structure data, annotations for biological data, results of complex computations, genetic sequences and multiple bio-datasets. However, the heterogeneity of these data, have created a need for research in resource integration and platform independent processing of investigative queries, involving heterogeneous data sources.

When processing huge amounts of data, information integration is one of the most critical issues, because it's crucial to preserve the intrinsic semantics of all the merged data sources. This integration would allow the proper organization of data, fostering the analysis and access the information to accomplish critical tasks, such as the processing of micro-array data to study protein function and medical researches in making detailed studies of protein structures to facilitate drug design (Ignacimuthu, 2005). Furthermore, DNA micro-array research community urgently requires technology to allow up-to-date micro-array data information to be found, accessed and delivered in a secure framework (Sinnot, 2007).

Several research disciplines, such as Bioinformatics, where information integration is critical, could benefit from harnessing the potential of a new approach: the Semantic Web (SW). The SW term was coined by Berners-Lee, Hendler and Lassila (2001) to describe the evolution of a Web that consisted of largely documents for humans to read towards a new paradigm that included data and information for computers to manipulate. The SW is about adding machine-understandable and machine-processable metadata to Web resource through its key-enabling technology: ontologies (Fensel, 2002). Ontologies are a formal explicit and shared specification of a conceptualization. The SW was conceived as a way to solve the need for data integration on the Web.

This article expounds SAMIDI, a Semantics-based Architecture for Micro-array Information and Data Integration. The most remarkable innovation offered by SAMIDI is the use of semantics as a tool for leveraging different vocabularies and terminologies and foster integration. SAMIDI is composed of a methodology for the unification of heterogeneous data sources from the analysis of the requirements of the unified data set and a software architecture.

BACKGROUND

This section introduces Bioinformatics and its need to process massive amounts of data; the benefit of the integration of the existing data sources of biological information and semantics, a tool for integration.

Bioinformatics

The term Bioinformatics was coined by Hwa Lim in the late 1980s, and later popularized through its association with the human genome project (Goodman, 2002). Bioinformatics is the application of information science and technologies for the management of biological data (Denn & MacMullen, 2002) and it describes any use of computers to store, compare, retrieve, analyze or predict the composition of the structure of biomolecules (Segall & Zhang, 2006). Research on Biology requires Bioinformatics to manipulate and discover new biological knowledge at several levels of increasing complexity. Biological data are produced through high-throughput methods (Vyas & Summers, 2005), which means that they have to be represented and stored in different formats, such as micro-arrays.

Micro-Array Data Sources

A DNA micro-array is a collection of microscopic DNA spots attached to a solid surface forming an array for the purpose of expression profiling, which monitors expression levels for thousands of genes simultaneously. Those features are read by a scanner that measures the level of activation, and the data is downloaded onto a computer for subsequent analysis (Cohen, 2005). Micro-arrays allow investigating million of genes simultaneously (Segall & Zhang, 2006). A biological experiment may require hundreds of micro-array, where a single micro-array generates up to million fragments of data (Murphy, 2002). This fact makes data analysis and management a major challenge for gene expression studies using micro-arrays (Xu, Maresh, Giardina & Pincus, 2004).

The need to manage data generated from Bioinformatics is crucial. Understanding biological processes necessitates access to collections of potentially distributed, separately owned and managed biological data sets (Sinnott, 2007). These data sources reside in different storages, hardware platforms, data base management systems, data models and data languages (Chen, Promparmote & Maire, 2006), which makes impossible their integration. To make things worse, this incompatibility is not limited to the use of different data technologies, but also because of its incompatibility in terms of semantics. This heterogeneity can be of two sorts: *syntactic* and *semantic* (Verschelde, Dos Santos, Deray, Smith & Ceusters, 2004). Syntactic

heterogeneity refers to differences in data models and data languages and can be easily resolved. Semantic heterogeneity refers to the underlying meanings of the data represented. It gives origin to naming conflicts and structural conflicts.

This incompatibility, and the necessity of sharing and aggregating information among the existing micro-array data sources leads researchers to seek for data integration.

Micro-Array Data Integration

Data analysis and management represent a major challenge for gene expression studies using micro-arrays (Xu, Maresh, Giardina & Pincus, 2004). Micro-array technology is still rather new and standards are not established (Murphy, 2002). This lack of standardization impedes micro-array data exchange. However, several projects have been started with a common goal: facilitate the exchange and analysis of micro-array data. MIAME (Minimum Information About Micro-array Experiment) is an XML based standard for the description of micro-array experiments. It's gaining importance because is required by numerous journals for the submission of articles providing micro-array experiments results. The purpose of MIAME is to define the core information needed for the description of an array based gene expression monitoring experiment. MAGE (Micro-Array Gene Expression) is a standard micro-array data model and exchange format that is able to capture information specified by MIAME.

Integration and Semantics

The ambiguity of terms, both within and between different databases and terminologies, makes integrating bioinformatics data task highly error prone (Verschelde *et al*, 2004). Converting all this information into a common data format will likely never be achieved and, therefore, the solution to the effective information management problem will necessarily go through the establishment of a common understanding. At this point is where semantics comes into play, bridging nomenclature and terminological inconsistencies to comprehend underlying meaning in a unified manner. The key elements that enable semantic interoperability are ontologies; semantic models of the data and they interweave human understanding of symbols with their machine processability (Della Valle, Cerizza, Bicer,

Kabak, Laleci & Lausen, 2005). Ontologies allow to organise terms used in heterogeneous data sources according to their semantic relationships so that heterogeneous data fragments can be mapped into a consistent frame of reference (Buttler, Coleman, Critchlow, Fileto, Han, Pu, Rocco & Xiong, 2002).

Applying semantics allows capturing the meaning of data one single time. Without semantics, each data element will have to be interpreted several times from its design and implementation until its use, facilitating error raise. Finally, semantics allows turning a great set of data sources into an integrated, coherent and unique body of information. The architecture of the information itself contains a record to keep the meaning and locate each data asset, enabling the automation of overlapping and redundancy analysis.

One of the most important contributions of ontology to the unification of biological data schemas is the MGED (Microarray Gene Expression Data) ontology (Whetzel, Parkinson, Causton, Fan, Fostel, Fragoso, Game, Heiskanen, Morrison, & Rocca-Serra, 2006) that was prompted by the heterogeneity among MI-AME and MAGE formats. MGED is a conceptual model for micro-array experiments that establishes concepts, definitions, terms and resources for standardized description of a micro-array experiment in support of MAGE. MGED has been accepted as a unifying terminology by the micro-array research community, which makes it a perfect candidate for becoming a universal understanding model.

THE SAMIDI APPROACH

SAMIDI is both a methodology to allow the conversion of a set of micro-array data sources into a unified representation of the information they include, and a software architecture. A general overview of SAMIDI is depicted in Figure 1.

The Unifying Information Model (UIM)

It brings together all the physical data schemas related to the data sources to be integrated. It doesn't replicate any of the data models, but is built to represent an agreed-upon scientific view and vocabulary which will be the foundation to understand the data. The UIM might capture the major concepts present in each schema for, after applying a semantic mapping, relate the physical schemas of the various data sources to the Model itself. Thus, the semantic mapping encapsulates the meaning of the data, taking into account the agreed-upon terminology. The UIM also provides the basis for the creation of new data assets, assuring they are consistent with the underlying schemas, and serves as a reliable reference for understanding the interrelationship between seemingly unrelated sources and for automatically planning the translation of data elements between them.

Figure 1. SAMIDI
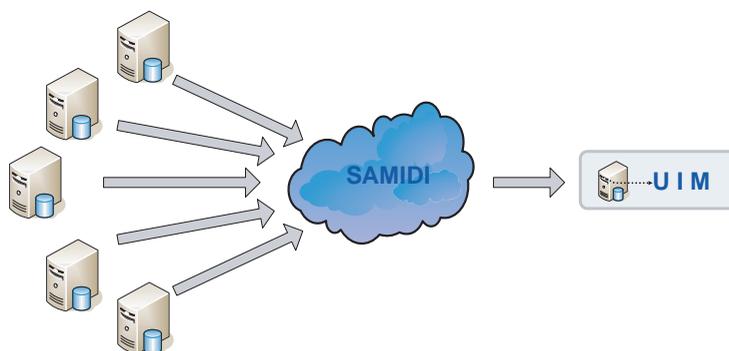
The Semantic Information Management Methodology (SIMM)

Its aim is to fill the existent gap between the fragmented information scenario represented by the set of micro-array data sources to be integrated and the required semantic data integration. The methodology is arranged in such a way that each of the phases generates a substantial added value itself, while concurrently progressing towards the full semantic integration. Next, the different stages of the methodology (Figure 2) are detailed:

1. *Capture Requirements*: The project scope is established by surveying the relevant data sources to be integrated and determining the information requirements of the aimed information model.
2. *Collect Metadata*: Data assets are classified and catalogued while the relevant (according to the organization's use of data) metadata are collected.
3. *Build UIM*: The structure of the UIM is determined by representing the desired business world-view, a comprehensive and consistent vocabulary and a set of business rules.
4. *Rationalize data semantics*: The meaning of the represented data is mapped into the UIM.
5. *Deploy*: The UIM, the metadata and the semantics are shared among the intended stakeholders and customized to fulfill their needs.
6. *Exploit*: Business processes are created, ensuring that the architecture has achieved the data management, data integration and data quality objectives.

The effective application of the SIMM requires a support information system. Key components of the supporting system should include:

- A repository for storing the collected metadata on data assets, schemas and models.
- A set of semantic tools for integrated ontology modeling for the creation of the UIM and the semantic mapping of the data schemas to the model.
- The standard business terminology stemmed from the UIM should be used across the supporting system.
- Data management capabilities of the system should include authoring and editing the information model; discovering data assets for any particular business concept; creating qualitative and quantitative reports about data assets; testing and simulating the performance of the UIM and impact analyzing in support of change.
- The system should fully support data integration by automatically generating code for queries and translation scripts between mapped schemas harnessing the common understanding acquired from the data semantics.
- Data quality should be approached by supporting the identification and decommissioning of redundant data assets, comparison mechanisms for ensuring consistency among semantically dissimilar data, and validation/cleansing of individual sources against the central repository of rules.
- The system should allow bi-directional communication with other systems for exchanging metadata and models by means of adaptors and

Figure 2. SIMM
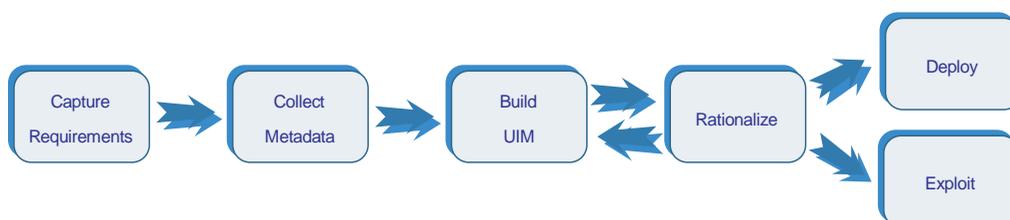
standards such as XML Metadata Interchange standard. Similarly, the system should be able to collect metadata and other data assets from relational databases or other kind of repositories.

- Capability of active data integration. The system should have a Run-Time interface for the automatic generation and exporting of queries, translation scripts, schemas and cleansing scripts.
- The user interface should provide a rich thick-client for power users in the data management group.
- The system should include a transversal platform supporting shared functionalities such as version control, collaboration tools, access control and configuration for all metadata and active content in the system.

The SAMIDI Software Architecture

A detailed description of the components comprised in the SAMIDI software architecture (see Figure 3) is presented now:

- SearchBot: Software agent whose mission is to perform a methodical, systematic and automatic browsing of the integrated information sources.
- Semantic Engine: Integrated set of tools for the semantic mapping of the data schemas to the

UIM using the MGED ontology. It will provide a semi-automatic mechanism for the mapping of schemas and concepts or categories in the UIM so as to lessen the workload of a process that will require human intervention. Fully automatic mapping has been discarded since it is regarded as not recommendable due to semantic incompatibilities and ambiguities among the different schemas and data formats. The objective of the Semantic Engine is to bridge the gap between cost-efficient machine-learning mapping techniques and pure human interaction.

- YARS: The YARS (Yet Another RDF Store) (Harth & Decker, 2005) system is a semantic data store that allows semantic querying and offers a higher abstraction layer to enable fast storage and retrieval of large amounts of metadata descriptions while keeping a small footprint and a lightweight architecture approach.
- GUI: Enables user-system interaction. It collects requests following certain search criteria, transfers the appropriate commands to and displays the results provided as a response from the RunTime Manager.
- Query Factory: Build queries into YARS storage systems using a query language. The semantics of the query are defined by an interpretation of the most suitable results of the query instead of

Figure 3. SAMIDI architecture
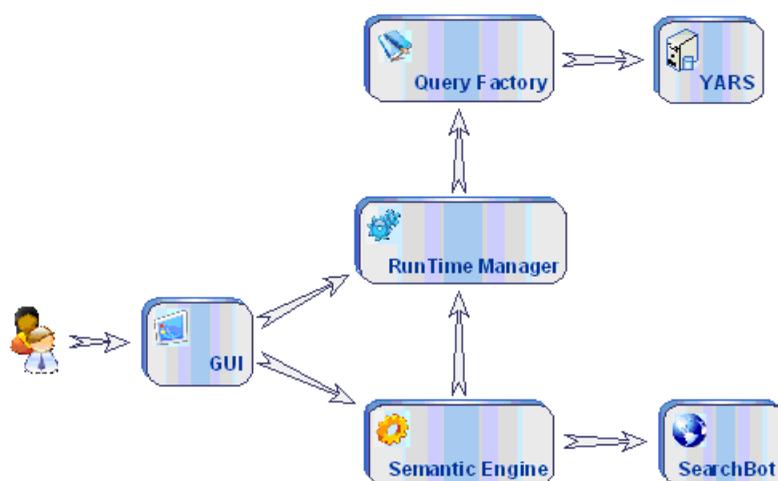
strictly rendering a formal syntax. YARS stores RDF triples, and the query factory, fro pragmatic reasons, implements SPARQL Query Language for RDF (Prud'hommeaux & Seaborne, 2004).

- RunTime Manager: This component coordinates the interactions among the rest of components. First of all, it communicates with the Semantic Engine to verify that the information collected by the SearchBot is adequately mapped on the MGED ontology as a UIM and stored into YARS using RDF syntax. Secondly, if accepts the users' search requests through the GUI and hands them over the Query Factory, which, in turn queries YARS to retrieve all the metadata descriptions related to the particular search criteria. By retrieving a huge amount of metadata information from all the integrated data sources, the user benefits from a knowledge aware search response which is mapped to the underlying terminology and unified criteria of the UIM, with the added advantage that all resources can be tracked and identified separately.

FUTURE TRENDS

We believe that the SW and SW Services (SWS) paradigm promise a new level of data and process integration that can be leveraged to develop novel high-performance data and process management systems for biological applications.

Using semantic technologies as a key technology for interoperation of various datasets enables data integration of the vast amount of biological and biomedical data. In a nutshell, the use of knowledge-oriented biomedical data integration would lead to achieving Intelligent Biomedical Data Integration, which will bring biomedical research to its full potential.

A future trend of SAMIDI effort is to integrate it in a SWS scenario in order to achieve seamless integration, also from the service or process integration perspective. This would enable the access to a number of heterogeneous data resources which are accessed via a Web Service interface and it would open the scope and goals of SAMIDI to a broader base. Promising integrating efforts in that direction have already been undertaken by the Biomedical Information and Integration Discovery with SWS (BIRD) platform (Gómez, Rico, García-Sánchez, Liu & Terra, 2007), which

fosters the intelligent interaction between natural language user intentions and the existing SWS execution environments. BIRD is a platform designed to interact with humans as a gateway or a man-in-the-middle towards SWS execution environments. The main goal of the system is to help users express their needs in terms of information retrieval and achieve information integration by means of SWS. BIRD allows users to state their needs via natural language or using a list of terms extracted from the Gene Ontology, infer the goals derived from the users' wishes and send them to the suitable SWS execution environment, which will retrieve the outcome resulting of the integration of the applications being accessed (e.g. all the biomedical publications and medical databases).

CONCLUSION

SAMIDI, represents a tailor-made contribution aimed to tackling the poser of discovering, searching and integrating multiple micro-array data sources harnessing the distinctive features of semantic technologies. The main contribution of SAMIDI is that it makes a decomposition of the unmanageable problem of integrating different and independent micro-array data sources.

SAMIDI is the first step to foster and extend the idea of using semantic technologies for the integration of different data sources not only originated from micro-array research but stemming from biomedical research areas.

REFERENCES

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The Semantic Web. *Scientific American Magazine.* 284 (5), 34-43.

Buttler, D., Coleman, M., Critchlow, T., Fileto, R., Han, W., Pu, C., Rocco, D. & Xiong, L. (2002). Querying multiple bioinformatics information sources: can semantic web research help? *ACM SIGMOD Record.* 31 (4), 59-64.

Chen, Y.P.P.,Promparmote, S. & Maire, F. (2006). MDSM: Microarray database schema matching using the Hungarian method. *Information Sciences.* 176 (19), 2771-2790.

Cohen, J. (2005). Computer Science and Bioinformatics. *Communitacions of the ACM.* 48 (3), 74-78.

Della Valle, E., Cerizza, D., Bicer, V., Kabak, Y., Laleci, G.B. & Lausen, H. (2005). The Need for Semantic Web Service in the eHealth. *W3C workshop on Frameworks for Semantics in Web Services.*

Denn, S.O. & MacMullen, W.J. (2002). The ambiguous bioinformatics domain: conceptual map of information science applications for molecular biology. *Proceedings of the 65th Annual Meeting of the American Society for Information Science & Technology.* Philadelphia. 18-21.

Gómez, J.M., Rico, M., García-Sánchez, F., Liu, Y., Terra, M. (2007). Biomedical Information Integration and Discovery with Semantic Web Services. *Proceedings of the 2nd International Work Conference on the Interplay between Natural and Artificial Computation.* 4527-4528.

Goodman, N. (2002). Biological data becomes computer literate: new advances in bioinformatics. *Current Opinion in Biotechnology.* 13 (1), 68-71.

Fensel, D. (2002). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce.* Springer-Verlag.

Harth, A. & Decker, S. (2005). Optimized Index Structures for Querying RDF from the Web. *Proceedings of the 3rd Latin American Web Congress.*

Ignacimuthu, S. (2005). *Basic Bioinformatics.* Alpha Science International.

Murphy, D. (2002). Gene expression studies using microarrays: principles, problems, and prospects. *Advances in Physiology Education.* 26 (4), 256-270.

Prud'hommeaux, E. & Seaborne, A. (2004). SPARQL Query Language for RDF. *WWW Consortium.*

Segall, R.S. & Zhang, Q. (2006). Data visualization and data mining of continuous numerical and discrete nominal-valued microarray databases for bioinformatics. *Kibernetes.* 35 (10), 1538-1566.

Sinnott, R.O. (2007). From access and integration to mining of secure genomic data sets across the Grid. *Future Generation Computer Systems.* 23 (3), 447-456.

Verschelde, J.L., Dos Santos, M.C., Deray, T., Smith, B. & Ceusters, W. (2004). Ontology-assisted database integration to support natural language processing and biomedical data-mining. *Journal of Integrative Bioinformatics.* 1 (1).

Vyas, H. & Summers, R. (2005). Interoperability of bioinformatics resources. *VINE: The journal of information and knowledge management systems.* 35 (3), 132-139.

Whetzel, P.L., Parkinson, H., Causton, H.C., Fan, L., Fostel, J., Fragoso, G., Game, L., Heiskanen, M., Morrison, N. & Rocca-Serra, P. (2006). The MGED Ontology: a resource for semantics-based description of microarray experiments. *Bioinformatics.* 22 (7), 866.

Xu, L., Maresh, G. A., Giardina, J. & Pincus, S. (2004). Comparison of different microarray data analysis programs and description of a database for microarray data management. *DNA & Cell Biology.* 23 (10), 643-652.

KEY TERMS

Bioinformatics: Application of information science and technologies for the management of biological data and the use of computers to store, compare, retrieve, analyze or predict the composition of the structure of biomolecules.

DNA Micro-Array: Collection of microscopic DNA spots attached to a solid surface forming an array for the purpose of expression profiling, which monitors expression levels for thousands of genes simultaneously.

MAGE: Standard micro-array data model and exchange format that is able to capture information specified by MIAME.

MGED: Ontology for micro-array experiments that establishes concepts, definitions, terms and resources for standardized description of a micro-array experiment in support of MAGE.

MIAME: XML based standard for the description of micro-array experiments, which stands for Minimum Information About a Micro-array Experiment.

M

Ontology: The specification of a conceptualization of a knowledge domain. It's a controlled vocabulary that describes objects and the relations among them in a formal way, and has a grammar for using the vocabulary terms to express something meaningful within a specified domain of interest.

Semantic Information Model Methodology: Set of activities, together with their inputs and outputs, aimed at the transformation of a collection of micro-array data sources into a semantically integrated and unified representation of the information stored in the data sources.

Unifying Information Model (UIM): Construction that brings together all the physical data schemas related to the data sources to be integrated. It's built to represent an agreed-upon scientific view and vocabulary which will be the foundation to understand the data.

Mobile Robots Navigation, Mapping, and Localization Part I

Lee Gim Hee
DSO National Laboratories, Singapore

Marcelo H. Ang Jr.
National University of Singapore, Singapore

INTRODUCTION

The development of autonomous mobile robots is continuously gaining importance particularly in the military for surveillance as well as in industry for inspection and material handling tasks. Another emerging market with enormous potential is mobile robots for entertainment.

A fundamental requirement for autonomous mobile robots in most of its applications is the ability to navigate from a point of origin to a given goal. The mobile robot must be able to generate a collision-free path that connects the point of origin and the given goal. Some of the key algorithms for mobile robot navigation will be discussed in this article.

BACKGROUND

Many algorithms were developed over the years for the autonomous navigation of mobile robots. These algorithms are generally classified into three different categories: *global path planners*, *local navigation methods* and *hybrid methods*, depending on the type of environment that the mobile robot operates within and the robot's knowledge of the environment.

In this article, some of the key algorithms for navigation of a mobile robot are reviewed. Advantages and disadvantages of these algorithms shall be discussed. The algorithms that are reviewed include the *navigation function*, *roadmaps*, *vector field histogram*, *artificial potential field*, *hybrid navigation* and the *integrated algorithm*. Note that all the navigation algorithms that are discussed in this article assume that the robot is operating in a planar environment.

GLOBAL PATH PLANNERS

Global path planning algorithms refer to a group of navigation algorithms that plans an optimal path from a point of origin to a given goal in a known environment. This group of algorithms requires the environment to be free from dynamic and unforeseen obstacles. In this section, two key global path planning algorithms: *navigation functions* and *roadmaps* will be discussed.

Navigation Functions

The most widely used global path planning algorithm is perhaps the navigation function computed from the "wave-front expansion" (J.-C Latombe, 1991; Howie Choset et al, 2005) algorithm due to its practicality, ease in implementation and robustness. The navigation function N is the *Manhattan distance* to the goal from the free space in the environment. The algorithm requires information of the environment provided to the robot to be represented as an array of grid cells.

Figure 1. The shaded cells are the 1-Neighbor of the cell (x, y) and the number shows the priority of the neighboring cells

2	3	4
1	(x, y)	5
8	7	6

Figure 2. Path generated by the navigation function

M
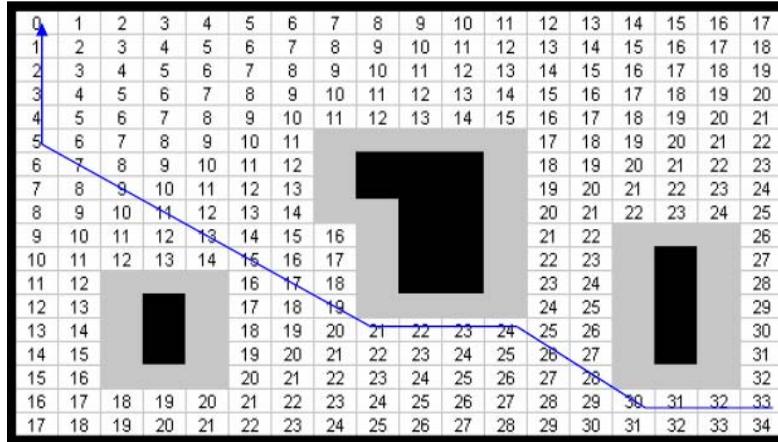
The navigation function assigns a numeric value *N* to each cell with the goal cell having the lowest value, and the other unoccupied cells having progressively higher values such that a steepest descent from any cell provides the path to the goal. The value of the unoccupied cell increases with the distance from the goal. Each grid cell is either free or occupied space denoted by gC_{free} and $gC_{occupied}$. First, the value of *N* is set to '0' at the goal cell gC_{goal}. Next, the value of *N* is set to '1' for every 1-Neighbor (see Figure 1 for the definition of 1-Neighbors) of gC_{goal} which is in gC_{free}. It is assumed that the distance between two 1-Neighbors is normalized to 1. In general, the value of each gC_{free} cell is set to *N*+1 (e.g., '2') for every unprocessed gC_{free} 1-Neighbor of the grid cell with value *N* (e.g., '1'). This is repeated until all the grid cells are processed.

Finally, a path to the goal is generated by following the steepest descent of the *N* values. To prevent the path from grazing the obstacles, the grid cells which are less than a safety distance α from the obstacles are omitted in the computation of the navigation function. Figure 2 shows a path generated by the navigation function. The black cells are the obstacles and the grey cells are the unsafe regions.

Roadmaps

A *roadmap* is a network of one-dimensional curves that captures the connectivity of free space in the en-

vironment (J.-C Latombe, 1991; Danner et al, 2000; Foskey et al, 2001; Isto P., 2002; T. Siméon et al, 2004; Xiaobing Zou et al, 2004; Howie Choset et al, 2005; Bhattacharya et al, 2007). Once a roadmap has been constructed, it is used as a set of standardized paths. Path planning is thus reduced to connecting the initial and goal positions to points in the roadmap. Various methods based on this general idea have been proposed. They include the *visibility graph* (Danner et al, 2000; Isto P., 2002; T. Siméon et al, 2004), *Voronoi diagram* (Foskey et al, 2001; Xiaobing Zou et al, 2004; Bhattacharya et al, 2007), *freeway net* and *silhouette* (J.-C Latombe, 1991; Howie Choset et al, 2005).

The *visibility graph* is the simplest form of *roadmap*. This algorithm assumes that the environment is made up of only polygonal obstacles. The nodes of a *visibility graph* include the point of origin, goal and all the vertices of the obstacles in the environment. The graph edges are straight line segments that connect any two nodes within the line-of-sight of each other. Finally, the shortest path from the start to goal can be obtained from the *visibility graph*.

Advantages and Disadvantages

The advantage of the *navigation functions*, *roadmaps* and other global path planning algorithms is that a continuous collision-free path can always be found by analyzing the connectivity of the free space. However,

these algorithms require the environment to be known and static. Any changes in the environment could invalidate the generated path. Hence, the navigation functions and other global path planning algorithms are usually not suitable for navigation in an initially unknown environment and those with dynamic and unforeseen obstacles.

LOCAL NAVIGATION METHODS

In contrast to the global path planners, local navigation methods do not require a known map of the environment to be provided to the robot. Instead, local navigation methods rely on current and local information from sensors to give a mobile robot online navigation capability. In this section, two of the key algorithms for local navigation: *artificial potential field* and *vector field histogram* will be evaluated.

Artificial Potential Field

The *artificial potential field* (O.Khatib, 1986) method, first introduced by Khatib, is perhaps the best known algorithm for local navigation of mobile robots due to its simplicity and effectiveness. The robot is represented as a particle in the configuration space q moving under the influence of an artificial potential produced by the goal configuration q_{goal} and the scalar distance to the obstacles. Typically the goal generates an attractive potential such as

$$U_g(q) = \frac{1}{2} K_g (q - q_g)^T (q - q_g)$$

(1)

which pulls the robot towards the goal, and each obstacle i produces repulsive potential such as

$$U_{i,o} = \begin{cases} \frac{1}{2} K_o \left(\frac{1}{d_i} - \frac{1}{d_o} \right)^2 & if \ d_i < d_0 \\ 0 & otherwise \end{cases}$$

(2)

which pushes the robot away from the obstacle. In cases where there is more than one obstacle, the total repulsive force is computed by the sum of all the repulsive forces produced by the obstacles. K_g and K_o are the respective gains of the attractive and repulsive potential. d_i is the scalar distance between the robot and obstacle i. The repulsive potential will only have effect on the robot when its moves to a distance which is lesser than d_0. This implies that d_0 is the minimum safe distance from the obstacle that the robot tries to maintain.

The negated gradient of the potential field gives the artificial force acting on the robot.

$$F(q) = -\nabla U(q)$$

(3)

Figure 3 shows the attractive force

$$F_g(q) = -K_g (q - q_g)$$

(4)
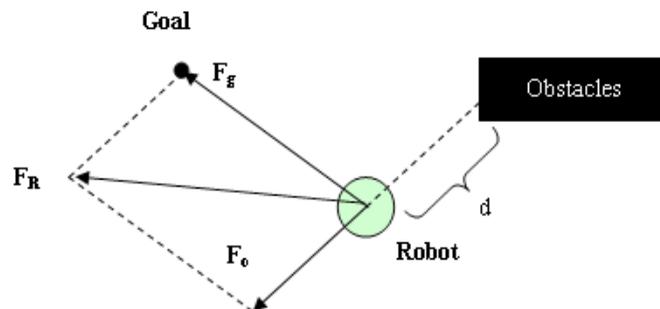
Figure 3. Robot's motion influenced by potential field

that is generated from the goal and the repulsive force

$$F_{i,o}(q) = \begin{cases} K_O \left(\dfrac{1}{d_i} - \dfrac{1}{d_O} \right) \dfrac{1}{d_i^2} & if \ d_i < d_O \\ \\ 0 & otherwise \end{cases} \quad (5)$$

that is generated from an obstacle i. F_R is the resultant of all the repulsive forces and attractive force. Note that n denotes the total number of obstacles which is lesser than a distance d_o from the robot. At every position, the direction of this force is considered as the most promising direction of motion for the robot.

$$F_R(q) = F_g(q) + \sum_{i=1}^{n} F_{i,o}(q) \quad (6)$$

Vector Field Histogram

The *vector field histogram* method (Y.Koren et al, 1991; J.Borenstien, 1991; Zhang Huiliang et al, 2003) requires the environment to be represented as a tessellation of grid cells. Each grid cell holds a numerical value that ranges from $0 - 15$. This value represents whether the environment represented by the grid cell is occupied or not. 0 indicates absolute certainty that the cell is not occupied and 15 indicates absolute certainty that the cell is occupied. A two stage data reduction process is carried out recursively to compute the desired motion of the robot at every instance of time.

In the first stage, the values of every grid cells that are in the vicinity of the robot's momentary location are reduced to a one-dimensional *polar histogram*. Each bin from the *polar histogram* corresponds to a direction as seen from the current location of the robot and it contains a value that represents the total sum of the grid cell values along that direction. The values from the *polar histogram* are also known as the *polar obstacle density* and they represent the presence of obstacles in the respective directions.

In the second stage, the robot selects the bin with a low *polar obstacle density* and direction closest to the goal. The robot moves in the direction represented by the chosen bin because this direction is free from obstacles and it will bring the robot closer to the goal.

Advantages and Disadvantages

The advantage of the *artificial potential field*, *vector field histogram* and other local navigation methods is that they do not include an initial processing step aimed at capturing the connectivity of the free space in a concise representation. Hence a prior knowledge of the environment is not needed. At any instant in time, the path is determined based on the immediate surrounding of the robot. This allows the robot to be able to avoid any dynamic obstacles in the robot's vicinity.

The major drawback of the local navigation methods is that they are basically steepest descent optimization methods. This renders the mobile robot to be susceptible to *local minima* (Y.Koren et al, 1991; J.-O. Kim et al, 1992; Liu Chengqing et al, 2000; Liu Chengqing, 2002; Min Gyu Park et al, 2004). A *local minimum* in the *potential field* method occurs when the attractive and the repulsive forces cancel out each other. The robot will be immobilized when it falls into a *local minimum*, and loses the capability to reach its goal. Many methods have been proposed to solve the *local minima* problem (J.-O. Kim et al, 1992; Liu Chengqing et al, 2000; Liu Chengqing, 2002; Min Gyu Park et al, 2004). For example, Liu Chengqing (Liu Chengqing et al, 2000; Liu Chengqing, 2002) has proposed the virtual obstacle method where the robot detects the *local minima* and fills the area with artificial obstacles. Consequently, the method closes all concave obstacles and thus avoiding *local minima* failures. Another method was proposed by Jin-Oh Kim (J.-O. Kim et al, 1992) to solve the *local minima* problem. This method uses *local minima free* harmonic functions based on fluid dynamics to build the artificial potentials for obstacle avoidances.

HYBRID METHODS

Another group of algorithms suggest a combination of the local navigation and global path planning methods. These algorithms aim to combine the advantages from both the local and global methods, and to also eliminate some of their weaknesses. In this section, two key hybrid methods algorithms: *hybrid navigation* and *integrated algorithm* will be reviewed.

M

Figure 4. Illustration of the hybrid navigation algorithm
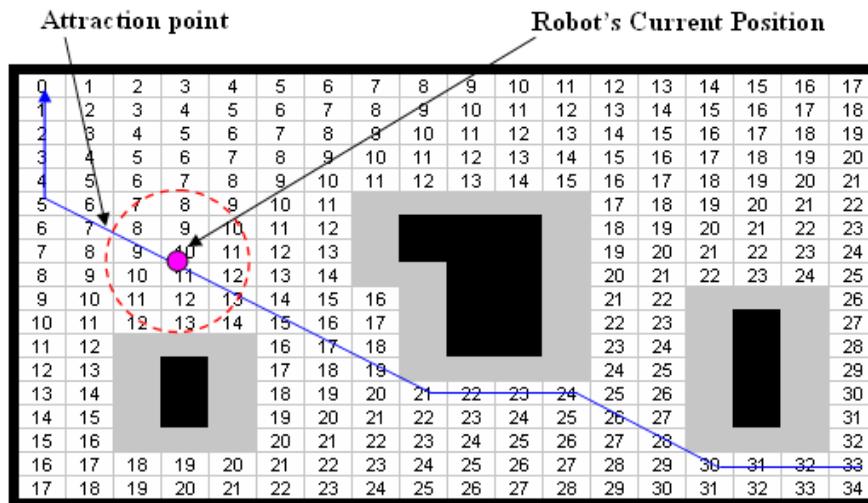
Hybrid Navigation

Figure 4 shows an illustration of the *hybrid navigation algorithm* (Lim Chee Wang, 2002; Lim Chee Wang et al, 2002). This algorithm combines the *navigation function* with the *potential field* method. It aims to eliminate local minima failures and at the same time does online collision avoidance with dynamic obstacles.

The robot first computes the path joining its current position to the goal using the *navigation function*. The robot then places a circle with an empirical radius centered at its current position. The cell that corresponds to the intersection of the circle with the *navigation function* path is known as the attraction point. The attraction point is the cell with the lowest N value if there is more than one intersection.

The robot advances towards the attraction point using the *potential field* method and the circle moves along with the robot which will cause the attraction point to change. As a result, the robot is always chasing after a dynamic attraction point which will progress towards the goal along the *local minima* free *navigation function* path. The radius of the circle is made larger to intersect the *navigation function* path in cases where no intersections are found. The radius of the circle is reduced to smaller than the distance between the robot and its

goal when the distance between the robot and its goal becomes smaller than the radius of the circle. This is to make sure that the N value of the next intersection will be smaller than the current N value.

Integrated Algorithm

In recent years, the *integrated algorithm* (Lee Gim Hee, 2005; Lee Gim Hee et al, 2007) has been proposed to give a mobile robot the ability to plan *local minima* free paths and does online collision avoidance to dynamic obstacles in an unknown environment. The algorithm modifies the *frontier-based exploration* method (Brian Yamauchi, 1997), which was originally used for map building, into a path planning algorithm in an unknown environment. The modified *frontier-based exploration* method is then combined with the *hybrid navigation* algorithm into a single framework.

Figure 5 shows an overview of the *integrated algorithm*. The robot first builds a local map (see Part II of this article for details on map building) of its surrounding. It then decides whether the goal is reachable based on the acquired local map. The goal is reachable if it is in free space, and is not reachable if it is in the unknown space. Note that an unknown region is a part of the map which has not been explored during the map building

M

Figure 5. The integrated algorithm
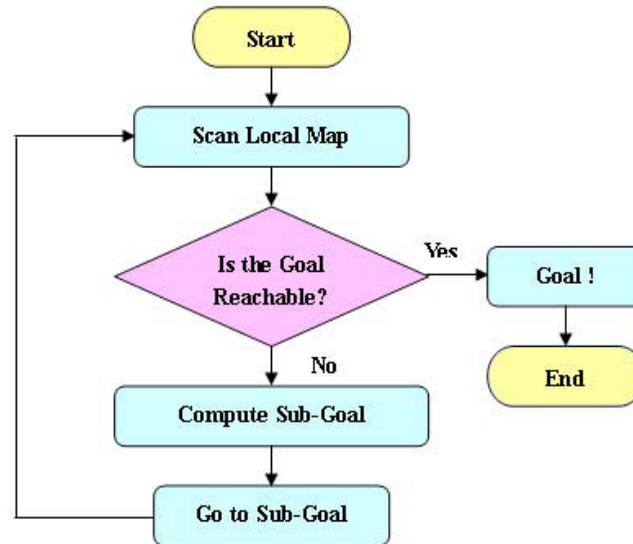
process. The robot will advance towards the goal using the *hybrid navigation* algorithm if it is reachable or advance towards the sub-goal and build another local map at the sub-goal if the goal is not reachable. This map will be added to the previous local maps to form a larger map of the environment. The process goes on until the robot finds the goal within a free space.

The sub-goal is computed in three steps. First, compute the path that joins the robot's current position and the goal using the *navigation function*. The unknown cells are taken to be free space in the computation of the *navigation function*. Second, all the frontiers in the map are computed. The boundary of free space and unknown region is known as the frontier. A frontier is made up of a group of adjacent frontier cells. The frontier cell is defined as any gC_{free} cell on the map with at least two unknown cells $gC_{unknown}$ as its immediate neighbor. The total number of frontier cells that make up a frontier must be larger than the size of the robot to make that frontier valid. Third, the frontier that intersects the navigation function path will be selected and its centroid chosen as the sub-goal.

Advantages and Disadvantages

The *hybrid navigation* algorithm has the advantage of eliminating local minima failures and at the same time doing online collision avoidance with dynamic obstacles. However, it requires the environment to be fully known for the search of a navigation function path to the goal. The algorithm will fail in a fully unknown environment. It also does not possess any capability to re-plan the *navigation function* path during an operation. Therefore any major changes to the environment could cause failure in the algorithm.

The integrated algorithm has the advantages of planning local minima free paths, does online collision avoidance to dynamic obstacles in totally unknown environments. In addition, the algorithm gives the mobile robot a higher level of autonomy since it does not depend on humans to provide a map of the environment. However, the advantages come with the trade off of tedious implementations. This is because the integrated algorithm requires both the *hybrid navigation* algorithm and a good mapping algorithm to be implemented at the same time.

CONCLUSION

Mobile robot navigation involves more than planning a path from a point of origin to a given goal. A mobile robot must be able to follow the planned path closely and avoid any dynamic or unforeseen obstacles during its journey to the goal. Some of the key algorithms that give a mobile robot navigation capability were discussed in this article. These algorithms include the *navigation function*, *roadmaps*, *artificial potential field*, *vector field histogram*, *hybrid navigation* and the *integrated algorithm*.

FUTURE TRENDS

The assumption that on-board sensors have perfect sensing capability is generally made by researchers researching on mobile robot navigation. In reality, these sensors are corrupted with noise and this usually causes adverse effects on the performance of the navigation algorithms. The greatest challenge for a robust implementation of the navigation algorithms is therefore to minimize the adverse effects caused by the sensor uncertainty.

REFERENCES

Bhattacharya, Priyadarshi, Gavrilova, & Marina L. (2007). Voronoi diagram in optimal path planning. 4th International Symposium on Voronoi Diagrams in Science and Engineering. pp. 38 - 47.

Brian Yamauchi. (1997). A frontier-based approach for autonomous exploration. Proceedings of the IEEE International Symposium on the Computational Intelligence in Robotics and Automation. pp. 146-151.

Danner, T. Kavraki, & L.E. (2000). Randomized planning for short inspection paths. IEEE International Conference on Robotics and Automation**. pp.** 971-976.

Foskey, M. Garber, M. Lin, & M.C. Manocha, D. (2001). A Voronoi-based hybrid motion planner. Proceedings of International Conference on Intelligent Robots and Systems. pp. 55-60.

Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Bugrad, Lydia E. Kavraki, & Sebastian Thrun. (2005). Principles of robot motion: Theory, algorithms, and implementations. MIT Press.

Isto P. (2002). Constructing probabilistic roadmaps with powerful local planning and path optimization. IEEE/RSJ International Conference on Intelligent Robots and System. Vol. 3, pp. 2323 - 2328.

J.Borenstien, & Y.Koren. (1991). The vector field histogram – Fast obstacle avoidance for mobile robots. IEEE Journal of Robotics and Automation. Vol 7, No 3, pp. 278-288.

J.-C Latombe. (1991). Robot motion planning. Kluwer Academic Publishers.

J.-O. Kim, & P. K. Khosla. (1992). Real-time obstacle avoidance using harmonic potential functions. Proceedings of IEEE Transactions on Robotics and Automation. Vol.8, pp. 338—349.

Lee Gim Hee, "Navigation of a mobile robot in an unknown environment ", Thesis for Bachelor of Engineering, National University of Singapore 2005.

Lee Gim Hee, Lim Chee Wang, & Marcelo H. Ang Jr. (2007). An integrated algorithm for autonomous navigation of a mobile robot in an unknown environment. Third Humanoid, Nanotechnology, Information Technology, Communication and Control Environment and Management (HNICEM) International Conference.

Lim Chee Wang. (2002). Motion planning for mobile robots. Thesis for Master of Engineering, National University of Singapore.

Lim Chee Wang, Lim Ser Yong, & Marcelo H. Ang Jr. (2002). Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment. Proceedings of the IEEE International Symposium on Intelligent Control. pp. 821-826.

Liu Chengqing. (2002). Sensor based local path planning for mobile robots. Master's Thesis, National University of Singapore.

Liu Chengqing, Marcelo H. Ang Jr, H. Krishnan, & Lim Ser Yong (2000). Virtual obstacle concept for local minima recovery in potential field based navigation.

Proceedings of the IEEE Conference on Robotics and Automation. Vol. 2, pp. 983-988.

Min Gyu Park , & Min Cheol Lee. (2004). Real-time path planning in unknown environment and a virtual hill concept to escape local minima. 30th Annual Conference of IEEE Industrial Electronics Society. Vol. 3, pp. 2223 - 2228**.**

O.Khatib. (1986). Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotic Research. Vol. 5, No. 1, pp.90-98.

T. Siméon, J.-P. Laumond, & C. Nissoux. (2004). Visibility-based probabilistic roadmaps for motion planning. Journal of Advanced Robotics, Brill Academic Publishers. pp. 477-493.

Xiaobing Zou, & Zixing Cai. (2004). Incremental environment modeling method based on approximate Voronoi diagram. Fifth World Congress on Intelligent Control and Automation. Vol.5, pp. 4618 - 4622.

Y.Koren, & J.Borenstien. (1991) Potential field methods and their inherent limitations for mobile robot navigation. Proceedings of the IEEE Conference on Robotics and Automation. pp.1398-1404.

Zhang Huiliang, & Huang Shell Ying. (2003). Dynamic map for obstacle avoidance. Proceedings of IEEE Intelligent Transportation Systems. Vol.2, pp. 1152 - 1157.

KEY TERMS

Global Path Planner: A group of navigation algorithms for planning an optimal path that connects a point of origin to a given goal in a known environment.

Graph Edge: Graph edge is usually drawn as a straight line in a graph to connect the nodes. It is used to represent connectivity between two or more nodes and may carry additional information such as the Euclidean distance between the nodes.

Graph Node: Graph Node is also known as graph vertex. It is a point on which the graph is defined and maybe connected by graph edges.

Hybrid Methods: A group of navigation methods that combine the global path planning and local navigation algorithms. The objective is to combine the advantages eliminate the inherent weaknesses of both groups of algorithms.

Local Minima: It is also known as relative minima. Local minimum refers to a minimum within some neighborhood and it may not be a global minimum.

Local Navigation Methods: A group of navigation algorithms that do not require a known map of the environment to be provided to the robot. Instead, local navigation methods rely on current and local information from sensors to give a mobile robot online navigation capability.

Manhattan Distance: The distance between two points measured along axes at right angles. For example, given two points p_1 and p_2 in a two-dimensional plane at (x_1, y_1) and (x_2, y_2) respectively, the Manhattan distance between p_1 and p_2 is given by $|x_1 - x_2| + |y_1 - y_2|$.

M

Mobile Robots Navigation, Mapping, and Localization Part II

Lee Gim Hee
DSO National Laboratories, Singapore

Marcelo H. Ang Jr.
National University of Singapore, Singapore

INTRODUCTION

In addition to the capability to navigate from a point of origin to a given goal and avoiding all static and dynamic obstacles, a mobile robot must posses another two competencies: *map building* and *localization* in order to be useful.

A mobile robot acquires information of its environment via the process of map building. Map building for mobile robots are commonly divided into *occupancy grid* and *topological* maps. *Occupancy-grid* maps seek to represent the geometric properties of the environment. *Occupancy-grid* mapping was first suggested by Elfes in 1987 and the idea was published in his Ph.D. thesis (A. Elfes, 1989) in 1989. *Topological* mapping was first introduced in 1985 as an alternative to the *occupancy-grid* mapping by R. Chatila and J.-P. Laumond (R. Chatila, & J.-P. Laumond, 1985). *Topological* maps describe the connectivity of different locations in the environment.

The pose of a mobile robot must be known at all times for it to navigation and build a map accurately. This is the problem of localization and it was first described in the late 1980's by R. Smith et al (R. Smith et al, 1980). Some key algorithms for map building and localization will be discussed in this article.

BACKGROUND

Map building is the process of acquiring information of the environment via sensory data and representing the acquired information in a format that is comprehensible to the robot. The acquired map of the environment can be used by the robot to improve its performance in navigation.

Localization is the process of finding the pose of the robot in the environment. It is perhaps the most important competency that a mobile robot must possess. This is because the robot must know its pose in the environment before it can plan its path to the goal or follow a planned path towards the goal.

In this article, two key algorithms for map building: *occupancy-grid* and *topological* mapping are discussed. The *occupancy grid* and *topological* maps are two different methodologies to represent the environment in a robot's memory. Two key localization methods: Localization with *Kalman filter* and *particle filter* are also reviewed.

MAP BUILDING

As seen from the *integrated algorithm* from part I of the article, a mobile robot must be able to acquire maps of an unknown environment to achieve higher level of autonomy. Map building is the process where sensory information of the surrounding is made comprehensive to a mobile robot. In this section, two key approaches for map building: *occupancy-grid* and *topological* mapping are discussed.

Occupancy-Grid Maps

Occupancy-grid maps (H.P. Moravec, 1988; H.P. Moravec et al, 1989; A. Elfes, 1987, A. Elfes, 1989; S. Thrun et al, 2005) represent the environment as a tessellation of grid cells. Each of the grid cells corresponds to an area in the physical environment and holds an occupancy value which indicates the probability of whether the cell is occupied or free. The occupancy value of the ith grid cell at current time t will be denoted by $p_{t,i}$. Note that $p_{t,i}$ must be within the range of 0 to 1 following the axioms of probability. $p_{t,i} = [0,0.5)$ indicates the confidence level of a cell being empty where 0 indicates absolute certainty that the cell is empty. $p_{t,i}$

M

= (0.5,1] indicates the confidence level of a cell being occupied where 1 indicates absolute certainty that the cell is occupied. $p_{t,i} = 0.5$ indicates that the cell is an unexplored area.

A robot does not have any knowledge of the world when it was first placed in an unknown environment. It is therefore intuitive to set $p_{t,i} = 0.5$ for all i at time $t = 0$. The map is updated via the *log odds* (S. Thrun et al, 2005) representation of occupancy. The advantage of *log odds* representation is that it can avoid numerical instabilities for probability near 0 or 1. The ith grid cell that intercepts the sensor line of sight is updated according to

$$l_{t,i} = l_{t-1,i} + l_{sensor} \qquad (1)$$

where $l_{t-1,i}$ is the *log odds* computed from the occupancy value of the cell at *t-1*.

$$l_{t-1,i} = \log \frac{p_{t-1,i}}{1 - p_{t-1,i}} \qquad (2)$$

$l_{sensor} = l_{occ}$ if the cell corresponds to the sensor measurement and $l_{sensor} = l_{free}$ if the range to the cell is shorter than the sensor measurement. The other cells in the map remain unchanged.

Figure 1(a) illustrates the update process for the map. The cell that corresponds to the sensor measurement is shaded black and all the cells that intercept the sensor measurement beam are shaded white. Figure 1(b) shows a case where the sensor measurement equals to maximum sensor range and $l_{sensor} = l_{free}$ for all cells that intercepts the sensor beam. This is because it is assumed that no obstacle is detected if the sensor measurement equals to maximum sensor range. l_{occ} and l_{free} are computed from

$$l_{occ} = \log \frac{p_{occ}}{1 - p_{occ}} \text{ and } l_{free} = \log \frac{p_{free}}{1 - p_{free}} \qquad (3)$$

where p_{occ} and p_{free} denote the probabilities of the sensor measurement correctly deducing whether a grid cell is occupied or empty. The two probabilities must add up to 1 and their values depend on the accuracy of the sensor. p_{occ} and p_{free} will have values closer to 1 and 0 for an accurate sensor. The values of p_{occ} and p_{free} have to be determined experimentally and remain constant in the map building process.

Figure 1. Updating an occupancy grid map (a) when an obstacle is detected (b) when a maximum range measurement is detected, i.e. it is assumed that in this case no obstacle is detected

(a)

(b)
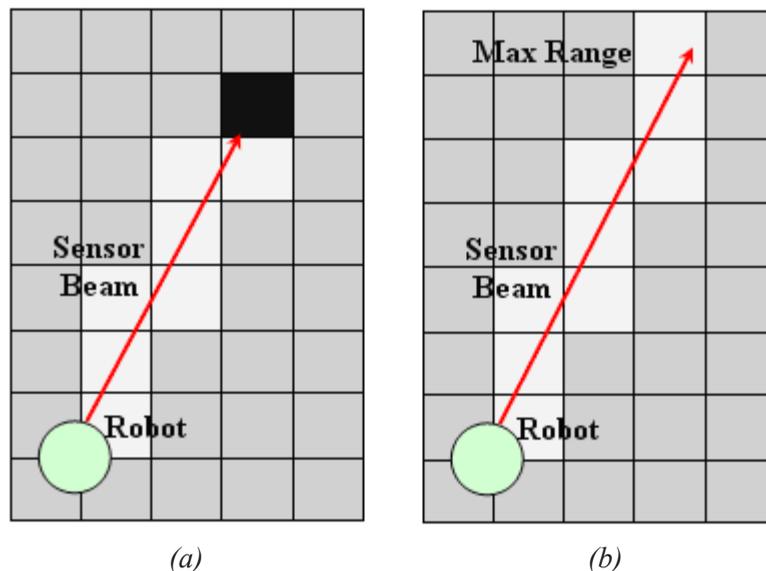
Figure 2. Occupancy grid map of the corridor along block EA level 3 in the Faculty of Engineering of the National University of Singapore (NUS)
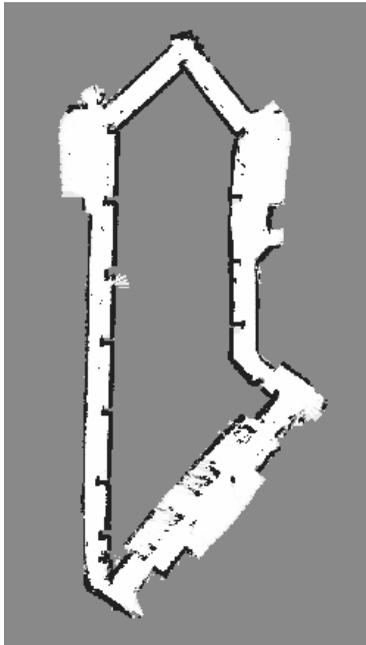
The occupancy value of a grid cell is easily recovered from

$$p_{t,i} = 1 - \frac{1}{1 + \exp\{l_{t,i}\}} \tag{4}$$

Figure 2 shows an occupancy grid map of the corridor along block EA level 3 in the Faculty of Engineering of the National University of Singapore (NUS) acquired with a laser range finder. The black regions denote obstacles, white regions denote free space and grey regions denote unexplored areas.

Topological Maps

Unlike the *occupancy grid* maps, topological maps (D. Kortenkamp et al, 1994; H. Choset, 1996; H. Choset et al, 1996) do not attempt to represent the geometric information of the environment. Instead, *topological maps represent the environments as graphs*. An example of the topological map is shown in Figure 3. List of significant features such as walls, corners, doors or corridors are represented as nodes m_i and connectivity

Figure 3. Example of a topological map. The features are represented as nodes m_i. The connectivity and distance between features are represented as edges u_{jk}
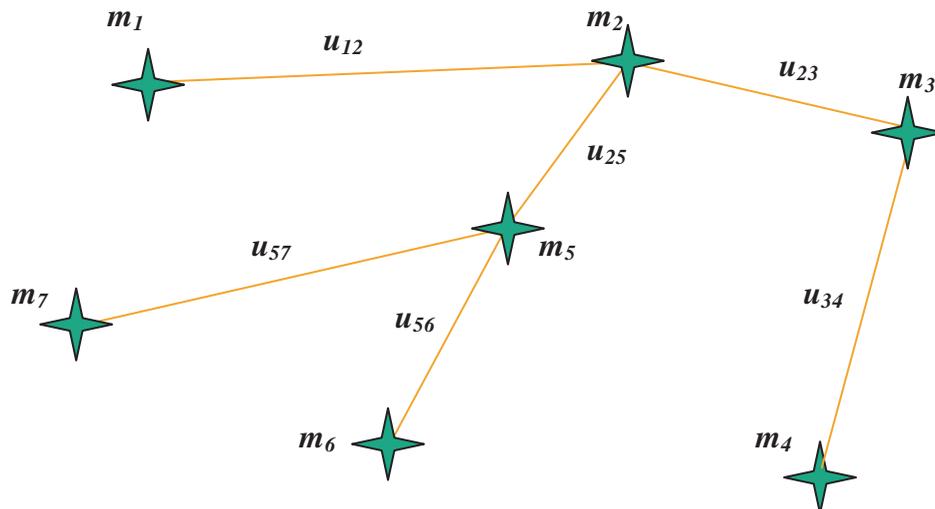
between adjacent features is represented as edges u_{jk}. In many *topological* maps, distances between adjacent features are also represented by the edges connecting the nodes. The success of the *topological* maps depends greatly on the efficiency in features extraction. Examples of feature extraction algorithms can be found in (Martin David Adams, 1999; Sen Zhang et al, 2003; Jodo Xavier et al, 2005).

Topological maps are better choice for mapping if memory space is a major concern. This is because less memory is required to store the nodes as compared to the large number of grid cells in occupancy grid maps. The advantage of less memory consumption for the *topological* map however comes with the tradeoff of being less accurate. This is because some important information such as precise location of the free spaces in the environment may not be represented in the maps. The limited accuracy of *topological* maps thus restricts the robot's capability for fast and safe navigation.

LOCALIZATION

Most mobile robots localize their pose x_t with respect to a given map based on odometry readings. Unfortunately, wheel slippages and drifts cause incremental localization errors (J. Borenstein et al, 1995; J. Borenstein et al, 1996). These errors cause the mobile robot to lose track of its own pose and hence losing the ability to navigate autonomously from one given point in the map to another. The solution to the localization problem is to make use of information of the environment from additional sensors. Examples of sensors used are laser range finder and sonar sensor that measure the distance between the robot and the nearest obstacles in the environment. The *extended Kalman filter* (EKF) and *particle filter* are two localization algorithms that use odometry and additional sensory data of the environment to localize a mobile robot. Both algorithms are probabilistic methods that allow uncertainties from the robot pose estimate and sensor readings to be accounted for in a principled way.

Localization with Extended Kalman Filter

EKF (John J. Leonard et al, 1991; A.Kelly, 1994; G. Welch et al, 1995; Martin David Adams, 1999; S. Thrun et al, 2005) is perhaps the most established algorithm for localization of mobile robots because of its robust-

ness and efficiency. The EKF is a recursive algorithm for estimating the pose of the robot with noisy sensor readings. A key feature of the EKF is that it maintains a posterior belief $bel(x_t)$ of the pose estimate, which follows a *Gaussian distribution*, represented by a mean x_t and covariance P_t. The mean x_t represents the most likely pose of the robot at time t and covariance P_t represents the error covariance of this estimate. The EKF consists of two steps: the prediction and update steps. In the prediction step, the predicted belief $\overline{bel}(x_t)$ is first computed using a motion model which describes the state dynamics of the robot. $\overline{bel}(x_t)$ is subsequently transformed into $bel(x_t)$ by incorporating the sensor measurements in the update step.

As mentioned above, the predicted belief $\overline{bel}(x_t)$ which is represented by the predicted mean $\overline{x}_t$ and covariance $\overline{P}_t$ is computed from the prediction step given by

$$\overline{x}_t = f(x_{t-1}, u_t) \tag{5}$$

$$\overline{P}_t = F_t P_{t-1} F_t^T + Q_t \tag{6}$$

where $f(.)$ is the motion model of the mobile robot, F is the Jacobian of $f(.)$ evaluated at x_{t-1}, Q_t is the covariance of the motion model and u_t is the control data of the robot.

$\overline{bel}(x_t)$ is subsequently transformed into $bel(x_t)$ by incorporating the sensor measurement z_t into the update step of the EKF shown in Equations 7, 8 and 9.

$$K_t = \overline{P}_t H_t^T (H_t \overline{P}_t H_t^T + R_t)^{-1} \tag{7}$$

$$x_t = \overline{x}_t + K_t (z_t - h(\overline{x}_t, m)) \tag{8}$$

$$P_t = (I - K_t H_t) \overline{P}_t \tag{9}$$

K_t, computed in Equation 7, is called the *Kalman gain*. It specifies the degree to which z_t should be incorporated into the new pose estimate. Equation 8 computes x_t by adjusting it in proportion to K_t and the deviation of the z_t with the predicted measurement $h(\overline{x}_t, m)$. It is important to note that the sensor measurement $z_t = [z_t^1 \quad z_t^2 \quad ...]^T$ refers to coordinates of a set of observed landmarks instead of the raw sensor readings and the sensor measurement model $h(.)$ gives the predicted measurement from the given topological map m and $\overline{x}_t$. H_t is the Jacobian of $h(.)$ evaluated at x_{t-1}. Finally, the covariance

P_t of the posterior belief $bel(x_t)$ is computed in Equation 9 by adjusting for the information gain resulting from the sensor measurements.

Localization with Particle Filter

In the recent years, there is an increasing interest in the use of particle filter (S. Thrun et al, 2001; C. Kwok et al, 2002; D. Fox et al, 2003; Ioannis M. Rekleitis, 2004; S. Thrun et al, 2005) over EKF for robot localization. This increased interest is likely due to four reasons. First, raw sensor measurements of the environment are used in particle filter localization where the EKF localization requires feature extraction. Second, the particle filter is more robust because unlike the EKF, it does not assume Gaussian distribution for the posterior belief $bel(x_t)$. Third, the particle filter is able to recover from localization failure. Localization failure occurs if the robot suddenly loses track of its pose during the localization process. Localization failure is also known as the kidnapped problem. Fourth, unlike the EKF there is no need to derive complicated Jacobians for the particle filter.

The intuition behind the particle filter is to represent the posterior belief $bel(x_t)$ by a finite sample set of M weighted particles. This sample set is drawn according to $bel(x_t)$. The particles set is denoted by

$$\xi_t = \chi_t^{[1]}, \chi_t^{[2]}, ..., \chi_t^{[M]} \tag{10}$$

where

$$\chi_t^{[m]} = [x_t^{[m]} \quad w_t^{[m]}]^T$$

denotes the mth particle. Here, $x_t^{[m]}$ is a random variable that represents a hypothesized state and $w_t^{[m]}$ is a non-negative value called the importance factor which represents the weight of each particle. Similar to the EKF, the particle filter consists of the prediction and update steps. In the prediction step, samples of the particles are drawn from a motion model of the robot to represent the predicted belief $\overline{bel}(x_t)$. The particles are then weighted according to the sensor measurements in the update step. Finally, $\overline{bel}(x_t)$ is transformed into the posterior belief $bel(x_t)$ by resampling the particles according to their weights.

Table 1 shows an iteration of the recursive particle filter algorithm for localization. The inputs to the particle filter are the set of particles representing the previous state belief ξ_{t-1}, the most recent control actions u_t and measurement data z_t. Line 3 is the prediction step that generates the hypothetical state $x_t^{[m]}$ by sampling from the motion model $p(x_t \mid u_t, x_{t-1}^{[m]})$ of the robot. The set of particles obtained after M iterations represents $\overline{bel}(x_t)$. Line 4 computes $w_t^{[m]}$ from the sensor measurement

Table 1. Pseudo algorithm for mobile robot localization with particle filter

1. $\overline{\xi}_t = \xi_t = \varnothing$;
2. for $m = 1$ to M do
3.   generate random sample of $x_t^{[m]}$ from $p(x_t \mid u_t, x_{t-1}^{[m]})$;
4. $w_t^{[m]} = p(z_t \mid x_t^{[m]}, m)$;
5. $\overline{\chi}_t^{[m]} = [x_t^{[m]} \quad w_t^{[m]}]^T$;
6. end;
7. for $m = 1$ to M do
8.   draw $\chi_t^{[m]}$ from $\overline{\xi}_t$ with probability proportional to $w_t^{[1]}, \quad w_t^{[2]}, ..., \quad w_t^{[M]}$;
9. end;

model. The importance factor accounts for the mismatch between $\overline{bel}(x_t)$ and $bel(x_t)$. Finally, the resampling process from line 7 to 9 draws with replacement M particles from the temporary set $\overline{\xi}_t$ with a probability proportional to the importance factors. The distribution of $\overline{bel}(x_t)$ is transformed into $bel(x_t)$ by incorporating the importance factors in the resampling process.

Figure 4(a) to (d) shows an implementation result of a robot localizing itself in a corridor. The particle set is initialized to the initial known pose of the robot show in Figure 4(a). The particles are initialized uniformly within a circle with radii 100mm and the initial position of the robot is taken as the center. The orientation of the particles is also initialized uniformly within ±5º to

Figure 4. Implementation of the particle filter to solve the localization problem. Notice that the error from the odometry grows as the robot travels a greater distance
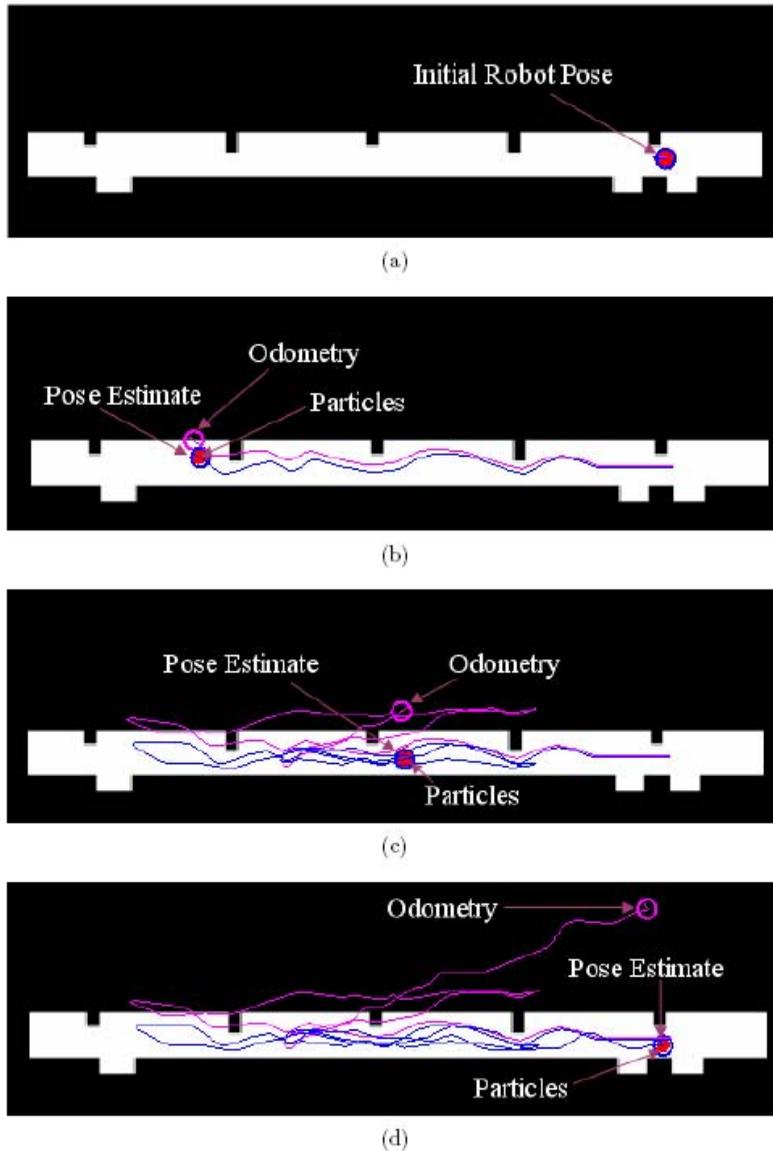
the initial orientation of the robot. This is to eliminate possible errors in estimating the initial pose of the robot. Figure 4(b) to 4(d) show that the error from the odometry grows as the robot travels a greater distance. The robot thinks that it is traveling in occupied space if it relied solely on the odometry readings and this is obviously wrong. It is apparent that the particle filter gives a more reasonable pose estimate because the robot is always moving within the free space.

It was mentioned earlier that the particle filter is able to recover from localization failure. An example of localization failure is when the robot is pushed by human resulting in a mismatch between the true and estimated pose of the robot. Fortunately, the problem can be easily solved by observing the total weights of the filter after each iteration. Localization failures will cause sharp drops in the total weights of the particles. The particles are re-initialized uniformly in the free space after detecting a sharp drop in the total weights of the particles. The particles will eventually converge to the true pose of the robot.

The particle filter is a powerful algorithm in solving the localization problem. However, it must be noted that the number of particles used to represent beliefs is an important parameter for efficiency of the particle filter in recovering from localization failures. A large size of particles is necessary to recover from localization failures in large environments and in many cases the maximum number particles is restricted by the available computing resources. This problem is also known as the *curse of dimensionality*.

CONCLUSION

A mobile robot has to possess three competencies to achieve full autonomy: navigation, map building and localization. Over the years, many algorithms have been proposed and implemented with notable success to give mobile robots all the three competencies. Some of the key algorithms such as the *navigation function*, *roadmaps*, *artificial potential field*, *vector field histogram*, *hybrid navigation* and the *integrated algorithm* for navigation; *occupancy grid* and *topological* based mapping; as well as the *Kalman filter* and *particle filter* for localization are reviewed in both Part I and II of this article.

FUTURE TRENDS

While the navigation, map building and localization algorithms are implemented with notable success, the scale and structure of the environments for these algorithms to work are limited. Hence, the future challenges for mobile robot autonomy are in the implementations of the algorithms in larger scale and more complex environments such as the urban cities or jungles.

REFERENCES

A. Elfes. (1987). Sonar-based real-world mapping and navigation. IEEE Journal of Robotics and Automation. RA-3(3):249–265.

A. Elfes. (1989). Occupancy grids: A probabilistic framework for robot perception and navigation. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University.

A.Kelly. (1994). A 3d state space formulation of a navigation Kalman filter for autonomous vehicle. Technical Report. The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213.

C. Kwok, D. Fox, & M. Meila. (2002). Real-time particle filters. Advances in Neural Information Processing Systems.

D. Fox, C. Kwok, & M. Meila. (2003). Adaptive real-time particle filters. Proceedings of the International Joint Conference on Robotics and Automation (ICRA).

D. Kortenkamp, & T.Weymouth. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. Proceedings of the Twelfth National Conference on Artificial Intelligence. pp. 979–984. AAAI Press/MIT Press.

G. Welch, & G.Bishop (1995). An introduction to the Kalman filter. Paper TR 95-041. Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175.

H. Choset. (1996). Sensor based motion planning: The hierarchical generalized voronoi graph. PhD thesis, California Institute of Technology.

H. Choset, & J.W. Burdick. (1996). Sensor based planning: The hierarchical generalized voronoi graph. Proceeding Workshop on Algorithmic Foundations of Robotics.

H.P. Moravec. (1988). Sensor fusion in certainty grids for mobile robots. AI Magazine. pp. 61-74.

H.P. Moravec, & D.W. Cho. (1989). A Bayesian method for certainty grids. AAAI 1989 Spring Symposium on Mobile Robots.

Ioannis M. Rekleitis. (2004). A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, 3480 University St., Montreal, Quebec, CANADA H3A 2A7.

J. Borenstein, & L. Feng. (1995). Umbmark: A benchmark test for measuring odometry errors in mobile robots. Proceedings of the 1995 SPIE Conference on Mobile Robots. pp. 569-574.

J. Borenstein, & L. Feng. (1996). Measurement and correction of systematic odometry errors in mobile robots. IEEE Transaction on Robotics and Automation. Vol. 12, pp. 869-880.

Jodo Xavier, Daniel Castrot, Marco Pachecot, Ant Nio Ruanot, & Urbano Nunes. (2005) Fast line, arc/circle and leg detection from laser scan data in a player driver. Proceedings of the 2005 IEEE International Conference on Robotics and Automation. pp. 95-106.

John J. Leonard, & Hugh F. Durrant-Whyte. (1991). Mobile robot localization by tracking geometric beacons. IEEE Transaction on Robotics and Automation. Vol. 7, pp. 376-382.

R. Chatila, & J.-P. Laumond. (1985). Position referencing and consistent world modeling for mobile robots. Proceedings of the 1985 IEEE International Conference on Robotics and Automation.

R. Smith, M. Self, & P. Cheeseman. (1990). Estimating uncertain spatial relationships in robotics. Autonomous Robot Vehicles. Springer-Verlag. pp. 167–193.

Martin David Adams. (1999). Sensor Modeling, Design and Data Processing for Autonomous Navigation. World Scientific. Vol. 13.

Sen Zhang, Lihua Xie, Martin Adams, & Fan Tang. (2003). Geometrical feature extraction using 2d range scanner. The Fourth International Conference on Control and Automation.

S. Thrun, D. Fox, W. Burgard, & F. Dellaert. (2001). Robust monte carlo localization for mobile robots. Artificial Intelligence Journal (AIJ).

S. Thrun, Wolfram Burgard, & Dieter Fox. (2005). Probabilistic robotics. Cambridge Massachusetts, London, England. The MIT Press.

KEY TERMS

Curse of Dimensionality: This term was first used by Richard Bellman. It refers to the problem of exponential increase in volume associated with adding extra dimensions to a mathematical space.

Gaussian Distribution: It is also known as normal distribution. It is a family of continuous probability distributions where each member of the family is described by two parameters: mean and variance. This form of distribution is used by the localization with *extended Kalman filter* algorithm to describe the posterior belief distribution of the robot pose.

Jacobians: The Jacobian is a first-order partial derivatives of a function. Its importance lies in the fact that it represents the best linear approximation to a differentiable function near a given point.

Odometry: A method to do position estimation for a wheeled vehicle during navigation by counting the number of revolutions taken by the wheels that are in contact with the ground.

Posterior Belief: It refers to the probability distribution of the robot pose estimate conditioned upon information such as control and sensor measurement data. The *extended Kalman filter* and *particle filter* are two different methods for computing the posterior belief.

Predicted Belief: It is also known as the prior belief. It refers to the probability distribution of the robot pose estimate interpreted from the known control data and in the absence of the sensor measurement data.

Recursive Algorithm: It refers to a type of computer function that is applied within its own definition. The

extended Kalman filter and *particle filter* are recursive algorithms because the outputs from the filters at the current time step are used as inputs in the next time step.

Modal Logics for Reasoning about Multiagent Systems

M

Nikolay V. Shilov
Russian Academy of Science, Institute of Informatics Systems, Russia

Natalia Garanina
Russian Academy of Science, Institute of Informatics Systems, Russia

INTRODUCTION

It becomes evident in recent years a surge of interest to applications of modal logics for specification and validation of complex systems. It holds in particular for combined logics of knowledge, time and actions for reasoning about multiagent systems (Dixon, Nalon & Fisher, 2004; Fagin, Halpern, Moses & Vardi, 1995; Halpern & Vardi, 1986; Halpern, van der Meyden & Vardi, 2004; van der Hoek & Wooldridge, 2002; Lomuscio, & Penczek, W., 2003; van der Meyden & Shilov, 1999; Shilov, Garanina & Choe, 2006; Wooldridge, 2002). In the next paragraph we explain what are logics of knowledge, time and actions from a viewpoint of mathematicians and philosophers. It provides us a historic perspective and a scientific context for these logics.

For mathematicians and philosophers logics of actions, time, and knowledge can be introduced in few sentences. A logic of actions (ex., Elementary Propositional Dynamic Logic (Harel, Kozen & Tiuryn, 2000)) is a polymodal variant of a basic modal logic **K** (Bull & Segerberg, 2001) to be interpreted over arbitrary Kripke models. A logic of time (ex., Linear Temporal Logic (Emerson, 1990)) is a modal logic with a number of modalities that correspond to "next time", "always", "sometimes", and "until" to be interpreted in Kripke models over partial orders (discrete linear orders for LTL in particular). Finally, a logic of knowledge or epistemic logic (ex., Propositional Logic of Knowledge (Fagin, Halpern, Moses & Vardi, 1995; Rescher, 2005)) is a polymodal variant of another basic modal logic **S5** (Bull & Segerberg, 2001) to be interpreted over Kripke models where all binary relations are equivalences.

BACKGROUND: MODAL LOGICS

All **modal logics** are languages that are characterized by syntax and semantics. Let us define below a very simple modal logic in this way. This logic is called **Elementary Propositional Dynamic Logic (EPDL)**.

Let *true, false* be Boolean constants, *Prp* and *Rel* be disjoint sets of propositional and relational variable respectively. The syntax of the classical propositional logic consists of formulas which are constructed from propositional variables and Boolean connectives "¬" (negation), "&" (conjunction), "∨" (disjunction), "→" (implication), and "↔" (equivalence) in accordance with the standard rules. EPDL has additional formula constructors, modalities, which are associated with relational variables: if r is a relational variable and φ is a formula of EPDL then

- *([r]φ)* is a formula which is read as "box r-φ" or "after r always φ";
- *(⟨r⟩φ)* is a formula which is read as "diamond r-φ" or "after r sometimes φ".

The semantics of EPDL is defined in models, which are called **labeled transition systems** by computer scientists and **Kripke models**[1] by mathematicians and philosophers. A model M is a pair *(D , I)* where the domain (or the universe) $D \neq \varnothing$ is a set, while the interpretation I is a pair of mappings *(P , R)*. Elements of the domain D are called states by computer scientists and worlds by mathematicians and philosophers. The interpretation maps propositional variables to sets of states $P: Prp \rightarrow 2^D$ and relational variables to binary relations on states $R: Rel \rightarrow 2^{D \times D}$. We write $I(p)$ and

I(r) instead of *P(p)* and *R(r)* whenever it is implicit that *p* and *r* are propositional and relational variables respectively.

Every model *M = (D , I)* can be viewed as a directed graph with nodes and edges labeled by propositional and action variables respectively. Its nodes are states of *D*. A node *s*∈*D* is marked by a propositional variable *p*∈*Prp* iff *s*∈*I(p)*. A pair of nodes *(s₁,s₂)*∈*D*×*D* is an edge of the graph iff *(s₁,s₂)*∈*I(r)* for some relational variable *r*∈*Rel*; in this case the edge *(s₁,s₂)* is marked by this relational variable *r*. Conversely, a graph with nodes and edges labeled by propositional and relational variables respectively can be considered as a model.

For every model *M = (D,I)* the **entailment** (validity, satisfiability) **relation** ⊨*M* between states and formulas can be defined by induction on formula structure:

- for every state *s* ⊨*M* *true* and not *s* ⊨*M* *false*;
- for any state *s* and propositional variable *p*, *s* ⊨*M* *p* iff *s*∈*I(p)*;
- for any state *s* and formula φ, *s* ⊨*M* *(¬φ)* iff it is not the case *s* ⊨*M* φ ;
- for any state *s* and formulas φ and ψ,
  s ⊨*M* *(φ &ψ)* iff *s* ⊨*M*φ and *s* ⊨*M*ψ ;
  s ⊨*M* *(φ ∨ψ)* iff *s* ⊨*M*φ or *s* ⊨*M*ψ ;
- for any state *s*, relational variable *r*, and formula φ,
  s ⊨*M* *([r]φ)* iff *(s,s′)*∈*I(r)* and *s′* ⊨*M* φ for every state *s′* ;
  s ⊨*M*⟨*r*⟩φ) iff *(s,s′)*∈*I(r)* and *s′* ⊨*M* φ for some state *s′* .

Semantics of the above kind is called **possible worlds semantics**.

Let us explain EPDL pragmatics by the following puzzle example.

Alice and Bob play the Number Game. Positions in the game are integers in [1..109]. An initial position is a random number. Alice and Bob make alternating moves: Alice, Bob, Alice, Bob, etc. Available moves are same for both: if a current position is n∈[1..99] then (n+1) and (n+10) are possible next positions. A player wins the game iff the opponent is the first to enter [100..109]. Problem: Find all initial positions where Alice has a winning strategy.

Kripke model for the game is quite obvious:

- States correspond to game positions, i.e. integers in *[1..109]*.
- Propositional variable *fail* is interpreted by *[100..109]*.
- Relational variable *move* is interpreted by possible moves.

Formula ¬*fail* & ⟨*move*⟩(¬*fail* & *[move]fail)* is valid in those states where the game is not lost, there exists a move after which the game is not lost, and then all possible moves always lead to a loss in the game. Hence this EPDL formula is valid in those states where Alice has a 1-round winning strategy against Bob.

COMBINING KNOWLEDGE, ACTIONS AND TIME

Logic of Knowledge

Logics of knowledge are also known as **epistemic logics**. One of the simplest epistemic logic is **Propositional Logic of Knowledge for *n>0* agents (PLK*ₙ*)** (Fagin, Halpern, Moses & Vardi, 1995). A special terminology, notation and **Kripke models** are used in this framework. A set of relational symbols *Rel* in PLK*ₙ* consists of natural numbers *[1..n]* representing names of agents. Notation for modalities is: if *i*∈ *[1..n]* and φ is a formula, then *(Ki φ)* and *(Si φ)* are used instead of *([i] φ)* and *(⟨i⟩ φ)*. These formulas are read as "(an agent) *i* knows φ" and "(an agent) *i* can suppose φ". For every agent *i*∈ *[1..n]* in every model *M = (D, I)*, interpretation *I(i)* is an "**indistinguishability relation**", i.e. an equivalence relation[2] between states that the agent *i* can not distinguish. Every model *M*, where all agents are interpreted in this way, is denoted as *(D, ~₁, … ~ₙ, I)* with explicit *I(1) = ~₁, … I(n) = ~ₙ* instead of brief standard notation *(D,I)*. An agent knows some "fact" φ in a state *s* of a model *M*, if the fact is valid in every state *s′* of this model that the agent can not distinguish from *s*:

- *s* ⊨*M* *(K*ᵢ *φ)* *iff s′* ⊨*M* φ for every state *s′*∼ᵢ *s*.

Similarly, an agent can suppose a "fact" φ in a state *s* of a model *M*, if the fact is valid in some state *s′* of this model that the agent can not distinguish from *s*:

- $s \models_M (S_i \varphi)$ *iff* $s' \models_M \varphi$ for some state $s' \sim_i s$.

The above possible worlds semantics of knowledge is due to pioneering research (Hintikka, 1962).

Temporal Logic with Actions

Another propositional polymodal logic is **Computational Tree Logic with actions (*Act-CTL*)**. *Act-CTL* is a variant of a basic propositional branching time temporal logic Computational Tree Logic (CTL) (Emerson, 1990; Clarke, Grumberg & Peled, 1999). In *Act-CTL* the set of relational symbols consists of action symbols *Act*. Each action symbol can be interpreted by an "instant action" that is executable in one undividable moment of time.

Act-CTL notation for basic modalities is: if $b \in Act$ and φ is a formula, then , $(A_b X \varphi)$ and $(E_b X \varphi)$ are used instead of $([b] \varphi)$ and $(\langle b \rangle \varphi)$. But syntax of *Act-CTL* has also some other special constructs associated with action symbols: if $b \in Act$ and φ and ψ are formulas, then $(A_b G \varphi)$, $(A_b F \varphi)$, $(E_b G \varphi)$, $(E_b F \varphi)$, $A_b(\varphi\ U\ \psi)$ and $E_b(\varphi\ U\ \psi)$ are also formulas of *Act-CTL*. In formulas of *Act-CTL* prefix "*A*" is read as "for every future", "*E*" – "for some future", suffix "*X*" – "next state", "*G*" – "always" or "globally", "*F*" – "sometimes" or "future", the infix "*U*" – "until", and a sub-index "*b*" is read as "in *b*-run(s)".

We have already explained semantics of $(A_b X \varphi)$ and $(E_b X \varphi)$ by referencing to $([b] \varphi)$ and $(\langle b \rangle \varphi)$. Constructs "$A_b G$", "$A_b F$", "$E_b G$", and "$E_b F$" can be expressed in terms of "$A_b(...U...)$" and "$E_b(...U...)$" , for example: $(E_b F \varphi) \leftrightarrow E_b(true\ U\ \varphi)$. Thus let us define below semantics of "$A_b(...U...)$" and "$E_b(...U...)$" only. Let $M = (D, I)$ be a model. If $b \in Act$ is an action symbol, then a partial *b*-run is a sequence of states $s_0, ...\ s_k, s_{(k+1)}, ... \in D$ (maybe infinite) such that $(s_k, s_{(k+1)}) \in I(b)$ for every consecutive pair of states within this sequence. If $b \in Act$ is an action symbol, then a *b*-run is an infinite partial *b*-run or finite *b*-run that can not be continued[3]. Then semantics of constructs "$A_b(...U...)$" and "$E_b(...U...)$" can be defined as follows:

- $s \models_M A_b(\varphi\ U\ \psi)$ iff for every *b*-run $s_0, ...s_k, ...$ that starts in s (i.e. $s_0=s$) there exists some $n \geq 0$ for which $s_n \models_M \psi$ and $s_k \models_M \varphi$ for every $k \in [0..(n-1)]$;
- $s \models_M E_b(\varphi\ U\ \psi)$ iff for some *b*-run $s_0, ...s_k, ...$ that starts in s (i.e. $s_0=s$) there exists some $n \geq 0$ for which $s_n \models_M \psi$ and $s_k \models_M \varphi$ for every $k \in [0..(n-1)]$.

The standard branching-time temporal logic *CTL* can be treated as *Act-CTL* with a single implicit action symbol.

Combined Logic of Knowledge, Actions and Time

There are many **combined polymodal logics** for reasoning about multiagent systems. Maybe the most advanced is **Belief-Desire-Intention (BDI) logic** (Wooldridge, 1996; Wooldridge, 2002). An agent's beliefs correspond to information the agent has about the world. (This information may be incomplete or incorrect. An agent's knowledge in BDI is just a true belief.) An agent's desires correspond to the allocated tasks. An agent's intentions represent desires that it has committed to achieving. Admissible actions are actions of individual agents; they may be constructed from primitive actions by means of composition, non-deterministic choice, iteration, and parallel execution. But semantics of BDI and reasoning in BDI are quite complicated for a short encyclopedia article.

In contrast, let us discuss below a simple example of a combined logic of knowledge, actions and time – namely **Propositional Logic of Knowledge and Branching Time for *n>0* agents *Act-CTL-K_n*** (Garanina, Kalinina, & Shilov, 2004; Shilov, Garanina & Choe, 2006; Shilov & Garanina, 2006). First we provide a formal definition of *Act-CTL-K_n*, then discuss some pragmatics, and then – in the next section – introduce model checking as a reasoning mechanism.

Let *[1..n]* be a set of agents ($n > 0$), and *Act* be a finite alphabet of action symbols. Syntax of *Act-CTL-K_n* admits epistemic modalities K_i , and S_i for every $i \in [1..n]$, and branching-time constructs $A_b X, E_b X, A_b G$, $E_b G$, $A_b F$, $E_b F$, $A_b(...U...)$, and $E_b(...U...)$ for every $b \in Act$. Semantics is defined in terms of entailment in environments. An **(epistemic) environment** is a tuple $E = (D, \sim_1, ... \sim_n, I)$ such that $(D, \sim_1, ... \sim_n)$ is a model for **PLK_n** , and (D, I) is a model for ***Act-CTL***. **Entailment relation** $\models$ is defined by induction according to the standard definition for propositional connectives (see semantics of ***EPDL***), and the above definitions of epistemic modalities and branching time constructs.

We are mostly interested in trace-based perfect recall synchronous environments generated from background finite environments. "Generated" means that possible "worlds" are runs of finite-state machine(s). There are several opportunities how to define semantics of

combined logics on runs. In particular, there are two extreme cases: **Forgetful Asynchronous Systems (FAS)** and **Synchronous systems with Perfect Recall (PRS)**. "Perfect recall" means that every agent has a log-file with all his/her observations along a run, while "forgetful" means that information of this kind is not available. "Synchronous" means that every agent can distinguish runs of different lengths, while "asynchronous" means that some runs of different lengths may be indistinguishable.

It is quite natural that in the FAS case combined logic *Act-CTL-K$_n$* can express as much as it can express in the background finite system. In contrast, in the PRS case *Act-CTL-K$_n$* becomes much more expressive than in the background finite environment. Importance of combined logics in the framework of trace-based semantics with synchronous perfect recall rely upon their characteristic as logics of agent's learning or knowledge acquisition. We would like to argue this characteristic by the following single-agent[4] Fake Coin Puzzle *FCP(N,M)*.

A set consists of (N+1) enumerated coins. The last coin is a valid one. A single coin with a number in [1..N] is fake, but other coins with numbers in [1...(N+1)] are valid. All valid coins have the same weight that differs from the weight of the fake. Is it possible to identify the fake by balancing coins M times at most?

In *FCP(N,M)* the agent (i.e. a person who have to solve the puzzle) does not know neither a number of the fake, nor whether it is lighter or heavier than the valid coins. Nevertheless, this number is in *[1..N]*, and the fake coin is either lighter (*l*) or heavier (*h*). The agent can make balancing queries and read balancing results after each query. Every balancing query is an action $b_{(L,R)}$ which consists in balancing of two disjoint sets of coins: with numbers $L\subseteq[1..N+1]$ on the left pan, and with numbers $R\subseteq[1..N+1]$ on the right pan, $|L| = |R|$. There are three possible balancing results: "<", ">", and "=", which means that the left pan is lighter, heavier than or equal to the right pan, respectively. Of course, there are initial states (marked by *ini*) which represent a situation when no query has been made.

Let us summarize. The agent acts in the environment generated from a finite space *[1..N]×{l,h}×{<, >, =, ini}*. His/her admissible actions are balancing query

$b_{(L,R)}$ for disjoint *L, R⊆ [1..N+1]* with *|L| = |R|*. The only information available for the agent (i.e., which gives him/her an opportunity to distinguish states) is a balancing result. The agent should learn *fake_coin_number* from a sequence which may start from any initial state and then consists of *M* queries and corresponding results. Hence single agent logic *Act-CTL-K$_1$* seems to be a very natural framework for expressing *FCP(N,M)* as follows: to validate or refute whether

$$s \models_E (E_B X ..._{M-times} ... E_B X (\vee_{f\in[1..N]}$$
$$K_1(fake\_coin\_number = f))...))$$

for every initial state *s*, where *E* is a PRS environment generated from a finite space *[1..N]×{l,h}×{<, >, =, ini}*, and *B* is a balancing query $\cup_{L,R\subseteq[1..N+1]} b_{(L,R)}$.

FUTURE TRENDS: MODEL CHECKING FOR COMBINED LOGICS

The **model checking problem** for a combined logic (*Act-CTL-K$_n$* in particular) and a class of **epistemic environments** (ex., PRS or FAS environments) is to validate or refute $s \models_E \varphi$, where *E* is a finitely-generated environment in the class, *s* is an "initial state" of the environment *E*, and φ is a formula of the logic. The above re-formulation of *FCP(N,M)* is a particular example of a **model checking problem** for a formula of *Act-CTL-K$_n$* and some finitely-generated perfect recall environment.

Papers (Meyden & Shilov, 1999) and (Garanina, Kalinina & Shilov, 2004) have demonstrated that if the number of agents *n>1*, then the model checking problem in perfect recall synchronous systems is very hard or even undecidable. In particular, it has non-elementary[5] upper and lower time bounds for *Act-CTL-K$_n$*. Papers (Meyden & Shilov, 1999) and (Shilov, Garanina & Choe, 2006) have suggested a tree-like data structures to make "feasible" model checking of combinations of temporal and action logics with propositional logic of knowledge *PLK$_n$*. Alternatively, (van der Hoek & Wooldridge, 2002; Lomuscio & Penczek, 2003) have susggested either to simplify language of logics to be combined, or to consider agents with "bounded" recall.

CONCLUSION

Combinations of temporal logics and logics of actions with logics of knowledge become an actual research topic due to the importance of study of interactions between knowledge and actions for reasoning about real-time multiagent systems. A comprehensive survey of logics, techniques, and results was out of scope of the article. The primary target of present article was to provide semi-formal introduction to the field of combined modal logics, discuss their utility for reasoning about multiagent systems. The emphasis has been done on model checking of trace-based knowledge-temporal specifications of perfect recall synchronous systems.

REFERENCES

Clarke, E., Grumberg, O., & Peled, D. (1999). *Model Checking*. MIT Press.

Bull, R. & Segerberg, K. (2001) Basic Modal Logic. *Handbook of Philosophical Logic*, v.3. D. Gabbay and F. Cuenthner editors. Kluwer Academic Publishers.

Dixon, C., Nalon, C., & Fisher, M. (2004). Tableau for Logics of Time and Knowledge with Interactions Relating to Synchrony. *Journal of Applied Non-Classical Logics*, 14(4), 397-445.

Emerson, E.A. (1990). Temporal and Modal Logic. *Handbook of Theoretical Computer Science (B)*, J. van Leeuwen, A.R. Meyer, M. Nivat, M. Paterson, D. Perrin editors. Elsevier and The MIT Press.

Fagin, R., Halpern, J.Y., Moses, Y., & Vardi, M.Y. (1995). *Reasoning about Knowledge*. MIT Press.

Garanina, N.O., Kalinina, N.A., & Shilov N.V. (2004) Model checking knowledge, actions and fixpoints. *Proceedings of Concurrency, Specification and Programming Workshop CS&P'2004*. Humboldt Universitat, Berlin, Informatik-Bericht, 170, 351-357.

Halpern, J.Y., & Vardi, M.Y. (1986). The Complexity of Reasoning About Knowledge and Time. *Proceedings of the eighteenth annual ACM symposium on Theory of computing*. 304-315.

Halpern, J. Y., van der Meyden, R., & Vardi, M.Y. (2004). Complete Axiomatizations for Reasoning about Knowledge and Time. *SIAM Journal on Computing*, 33(3), 674-703.

Harel, D., Kozen, D., & Tiuryn, J. (2000). *Dynamic Logic*. MIT Press.

Hintikka, J. (1962). *Knowledge and Belief.* Cornell University Press.

van der Hoek, W., & Wooldridge, M.J. (2002). Model Checking Knowledge and Time. *Lecture Notes in Computer Science*, 2318, 95-111.

Lomuscio, A., & Penczek, W. (2003). Verifying Epistemic Properties of Multi-agent Systems via Bounded Model Checking. *Fundamenta Informaticae*, 55(2), 167-185.

van der Meyden, R., & Shilov, N.V. (1999). Model Checking Knowledge and Time in Systems with Perfect Recall. *Lecture Notes in Computer Science*, 1738, 432-445.

Rescher, N. (2005). *Epistemic Logic. A Survey of the Logic of Knowledge*. University of Pitsburgh Press.

Shilov, N.V., Garanina, N.O., & Choe, K.-M. (2006). Update and Abstraction in Model Checking of Knowledge and Branching Time. *Fundameta Informaticae*, 72(1-3), 347-361.

Wooldridge, M. (1996) Practical reasoning with procedural knowledge: A logic of BDI agents with know-how. *Lecture Notes in Artificial Intelligence*, (1085), 663–678.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley & Sons Ltd.

KEY TERMS

Environment: A labeled transition system that provides an interpretation for logic of knowledge, actions and time simultaneously.

Labeled Transition Systems or Kripke Model: An oriented labeled graph (infinite maybe). Nodes of the graph are called states or worlds, some of them are marked by propositional symbols that are interpreted to be valid in these nodes. Edges of the graph are marked by relational symbols that are interpreted by these edges.

Logic of Actions: A polymodal logic that associate modalities like "always" and "sometimes" with action symbols that are to be interpreted in labeled transition systems by transitions. A so-called Elementary Propositional Dynamic Logic (EPDL) is sample logic of actions.

Logic of Knowledge or Epistemic Logic: A polymodal logic that associate modalities like "know" and "suppose" with enumerated agents or groups of agents. Agents are to be interpreted in labeled transition systems by equivalence "indistinguishability" relations. A so-called Propositional Logic of Knowledge of n agents (PLK$_n$) is sample epistemic logic.

Logic of Time or Temporal Logic: A polymodal logic with a number of modalities that correspond to "next time", "always", "sometimes", and "until" to be interpreted in labeled transition systems over discrete partial orders. For example, Linear Temporal Logic (LTL) is interpreted over linear orders.

Model Checking Problem: An algorithmic problem to validate or refute a property (presented by a formula) in a state of a model (from a class of Kripke structures). For example, model checking problem for combined logic of knowledge, actions and time in initial states of perfect recall finitely generated environments.

Multiagent System: A collection of communicating and collaborating agents, where every agent have some knowledge, intensions, enabilities, and possible actions.

Perfect Recall Synchronous Environment: An environment for modeling a behavior of a perfect recall synchronous system.

Perfect Recall Synchronous System: A multiagent system where every agent always records his/her observation at all moments of time while system runs.

ENDNOTES

[1] Due to pioneering papers of Saul Aaron Kripke (born in 1940) on models for modal logics.

[2] A symmetric, reflexive, and transitive binary relation on *D*.

[3] That is for the last state *s* there is no state *s′* such that $(s,s′) \in I(b)$.

[4] For multiagent example refer Muddy Children Puzzle (Fagin, Halpern, Moses & Vardi, 1995).

[5] I.e. it is not bounded by a tower of exponents with any fixed height

Modularity in Artificial Neural Networks

Ricardo Téllez
Technical University of Catalonia, Spain

Cecilio Angulo
Technical University of Catalonia, Spain

INTRODUCTION

The concept of modularity is a main concern for the generation of artificially intelligent systems. Modularity is an ubiquitous organization principle found everywhere in natural and artificial complex systems (Callebaut, 2005). Evidences from biological and philosophical points of view (Caelli and Wen, 1999) (Fodor, 1983), indicate that modularity is a requisite for complex intelligent behaviour. Besides, from an engineering point of view, modularity seems to be the only way for the construction of complex structures. Hence, whether complex neural programs for complex agents are desired, modularity is required.

This article introduces the concepts of modularity and module from a computational point of view, and how they apply to the generation of neural programs based on modules. Two levels, strategic and tactical, at which modularity can be implemented, are identified. How they work and how they can be combined for the generation of a completely modular controller for a neural network based agent is presented.

BACKGROUND

When designing a controller for an agent, there exists two main approaches: a single module contains all the agent's required behaviours (monolithic approach), or global behaviour is decomposed into a set of simpler sub-behaviours, each one implemented by one module (modular approach). Monolithic controllers implement on a single module all the required mappings between the agent's inputs and outputs. As an advantage, it is not required to identify required sub-behaviours nor relations between them. As a drawback, whether the complexity of the controller is high, it could be impossible at practice to design such a controller without obtaining large interferences between different parts of it. Instead, when a modular controller is used, the global controller is designed by a group of sub-controllers, so required sub-controllers and their interactions for generating the final global output must be defined.

Despite the disadvantages of the modular approach (Boers, 1992), complex behaviour cannot be achieved without some degree of modularity (Azam, 2000). Modular controllers allow the acquisition of new knowledge without forgetting previously acquired one, which represents a big problem for monolithic controllers when the number of required knowledge rules to be learned is large (De Jong et al., 2004). They also minimize the effects of the credit assignment problem, where the learning mechanism must provide a learning signal based on the current performance of the controller. This learning signal must be used to modify the controller parameters which will improve the controller behaviour. In large controllers, it becomes difficult finding changing parameters of the controller based on the global learning signal. Modularization helps to keep small the controllers' size, minimizing the effect of the credit assignment.

Modular approaches allow for a complexity reduction of the task to be solved (De Jong et al., 2004). While in a monolithic system the optimization of variables is performed at the same time, resulting in a large optimization space, in modular systems, optimization is performed independently for each module resulting on reduced searching spaces. Modular systems are scalable, in the sense that former modules can be used for the generation of new ones when problems are more complex, or just new modules can be added to the already existing ones. It also implies that modular systems are robust, since the damage on one module results in a loss of the abilities given by that module, but the whole system is partially kept functioning. Modularity can be a solution to the problem of neural interference (Di Ferdinando et al., 2000), which is encountered in monolithic networks. This phenomenon

is produced when an already trained network losses part of its knowledge when either, it is re-trained to perform a different task, called temporal cross-talk (Jacobs et al.,1991), or two or more different tasks at the same time, the effect being called spatial cross-talk (Jacobs,1990). Modular systems allow reusing modules in different activities, without re-implementation of the function represented on each different task (De Jong et al., 2004) (Garibay et al., 2004).

Modularity

From a computational point of view, modularity is understood as the property that some complex computational tasks have to be divided into simpler subtasks. Then, each of those simpler subtasks is performed by a specialized computational system called a module, generating the solution of the complex task from the solution of the simpler subtask modules (Azam, 2000). From a mathematical point of view, modularity is based on the idea of a system subset of variables which may be optimized independently of the other system variables (De Jong et al., 2004). In any case, the use of modularity implies that a structure exists in the problem to be solved.

In modular systems, each of the system modules operates primarily according to its own intrinsically determined principles. Modules within the whole system are tightly integrated but independent from other modules following their own implementations. They have either distinct or the same inputs, but they generate their own response. When the interactions between modules are weak and modules act independently from each other, the modular system is called nearly decomposable (Simon, 1969). Other authors have identified this type of modular systems as separable problems (Watson et al., 1998). This is by far one of the most studied types of modularity, and it can be found everywhere from business to biological systems. In nearly decomposable modular systems, the final optimal solution of a global task is obtained as a combination of the optimal solutions of the simpler ones (the modules).

However, the existence of decomposition for a problem doesn't imply that sub-problems are completely independent from each other. In fact, a system may be modular and still having interdependencies between modules. It is defined a decomposable problem as a problem that can be decomposed on other sub-prob-

lems, but the optimal solution of one of those problems depends on the optimal solution of some of the others (Watson, 2002). The resolution of such modular systems is more difficult than a typical separable modular system and it is usually treated as a monolithic one in the literature.

Module

Most of the works that use modularity, use the definition of module given by (Fodor, 1983), which is very similar to the concept of object in object-oriented programming: a module is a domain specific processing element, which is autonomous and cannot influence the internal working of other modules. A module can influence another only by its output, this is, the result of its computation. Modules do not know about a global problem to solve or global tasks to accomplish, and are specific stimulus driven. The final response of a modular system to the resolution of a global task, is given by the integration of the responses of the different modules by a especial unit. The global architecture of the system defines how this integration is performed. The integration unit must decide how to combine the outputs of the modules, to produce the final answer of the system, and it is not allowed to feed information back into the modules.

MODULAR NEURAL NETWORKS

When modularity is applied for the design of a modular neural network (MNN) based controller, three general steps are commonly observed: task decomposition, training and multi-module decision-making (Auda and Kamel, 1999). Task decomposition is about dividing the required controller into several sub-controllers, and assigning each sub-controller to one neural module. Modules should be trained either, in parallel, or in different processes following a sequence indicated by the modular design. Finally, when the modules have been prepared, a multi-module decision making strategy is implemented which indicates how all those modules should interact in order to generate the global controller response. This modularization approach can be seen as at the level of the task.

The previous general steps for modularity only apply for a modularization of nearly decomposable or separable problems. Decomposable problems, those

where strong interdependencies between modules exist, are not considered under that decomposition mechanism, and they are treated as monolithic ones. The article introduces the differentiation between two modular levels, the current modularization level, which concentrates on task sub-division, and a new modularization performed at the level of the devices or elements. Those approaches are called strategic and tactical, respectively.

Strategic and Tactical Modularity

Borrowing the concepts from game theory, strategy deals with what has to be done in a given situation in order to perform a task by dividing the global target solution into all the sub-targets required to accomplish the global one. Tactics, on the other hand, treats about how plans are going to be implemented, this means, how to use the resources available at that moment to accomplish each of those sub-targets.

It is defined strategic modularity in neural controllers as the modular approach that identifies which sub-goals are required for an agent in order to solve a global problem. Each sub-goal identified is implemented by a monolithic neural net. In contrast, tactical modularity in neural controllers is defined as the one that identifies which inputs and outputs are necessary for the implementation of a given goal, and it designs a single module for each input and output. In tactical modularity, modularization is performed at the level of the elements (any meaningful input or output of the neural controller) that are actually involved in the accomplishment of the task.

To our extent, all the research based on neural modularity and divide-and-conquer principles, focus their division at the strategic level, that is, how to divide the global problem into its sub-goals. Then, they implement each of those sub-goals by means of a single neural controller, final goal being generated by combining the outputs of those sub-goals in some sense. The current paper proposes, first, the definition of two different levels of modularity, and second, the use of tactical modularity as a new level of modularization that allocates space for decomposable modularity. It is expected that tactical modularization will be able in the generation of complex neural controllers when many inputs and outputs must be taken into account. It will be confirmed below, where the use of the two

types of modularity will be compared against monolithic approaches.

Implementing Modularity

Strategic modularity can be implemented by any of the modular approaches that already exist in the literature. See (Auda and Kamel, 1999) for a complete description. Any of the modularization methods described there is strategic, although it was not given that name, and they can, in general, be integrated with tactical modularity.

The term strategic is used for those modular approaches in order to differentiate them from the new proposed modularity.

Tactical modularity defines modularity at the level of the elements involved in the generation of a sub-goal. By elements, it is understand the inputs required to generate the sub-goal and the outputs that define the sub-goal solution. Each of those elements conform a tactical module, implemented by a simple neural network. That is, tactical modularity is implemented by designing a completely distributed controller composed of small processing modules around each of the meaningful elements of the problem.

The schematics for a tactical module is shown in Figure 1. Tactical modules are connected to its associated element, controlling it, and processing the information coming in, for input elements, or going out, for output elements. This kind of connectivity means that the processing element is the one that decides which commands must be sent to the output element, or how a value received from an input element must be interpreted. It is said that the processing element is responsible for its associated element.

In order to generate a complete answer for the sub-goal, all the tactical modules are connected each other, output of each module being sent back to all the others. By introducing this connectivity, each module is aware about what the others are doing, allowing that the different modules coordinate for the generation of a common answer, and avoiding a central coordinator. The resulting architecture shows a completely distributed MNN, where neural modules are independent but implement strong interactions with the other modules. Figure 2 shows an example of connectivity in the generation of a tactical modular neural controller for a simple system composed of two input elements and two outputs.

Figure 1. Schematics of a tactical module for one input element (left) and for one output element (right)
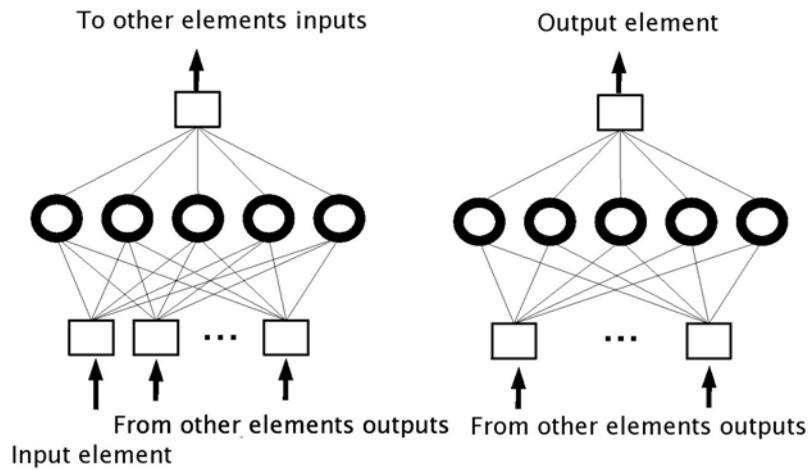
Figure 2. Connectivity of a tactical modular controller with two input elements and two output elements
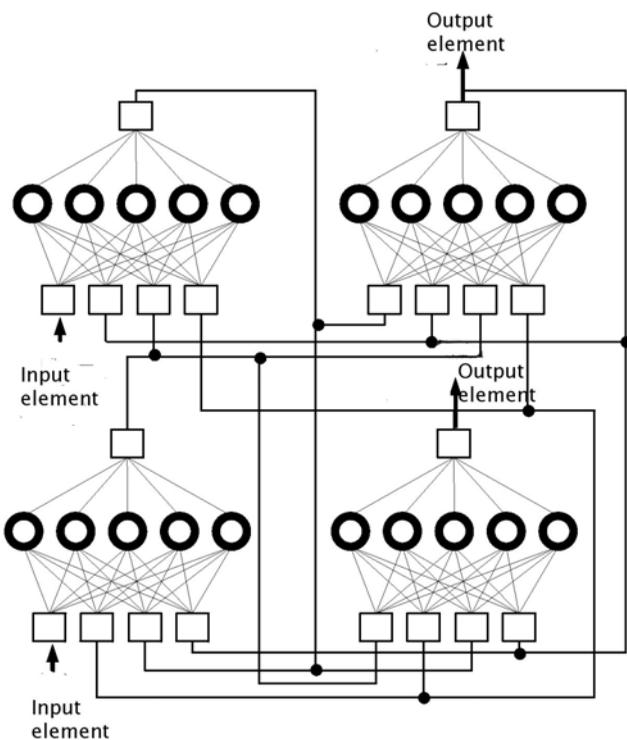
Training tactical modules is difficult due to the strong relationships between the different modules. Training methods used in strategic modules based on error propagation are not suitable, so a genetic algorithm is used to train the nets, because it allows to find the networks weights without defining an error measurement, just by specifying a cost function (Di Ferdinando et al., 2000).

Combination of Different Levels

The use of one kind of modularity does not prevent, in principle, the use at the same time of the other type of modularity. In fact, strategic and tactical modularity can be used separately or in conjunction with each other. When the solution required from the controller is simple, then either, a strategic, or a tactical modularization can be used. In those cases, it is suggested that the selection of the kind of modularity be based on the complexity of the problem. For simple problems with a small number of elements, a monolithic controller will fit it. Whether the number of elements is high, then a tactical modular controller will be the best option. Finally, for very complex tasks with many elements, a combination of strategic and tactical modularization could be preferable.

When combining both levels in one neural controller, the strategic modularization should be first performed, for identifying the different sub-goals required for the implementation. Next, a tactical modularization should be completed, implementing each of those sub-goals by a group of tactical modules. The number of tactical modules for each strategic module will depend on the elements involved in the resolution of the specific sub-goal.

Application Examples

So far, strategic and tactical modularity have been mainly applied to robot control. The input elements are sensors and the output elements are actuators. In a first experiment, tactical modularity was applied to the control of a Khepera robot learning to solve the garbage collector problem (Téllez and Angulo, 2006) (Téllez and Angulo, 2007). It involved the coordination of 11 elements (seven sensors and four actuators), creating 11 tactical modules. The task was compared with different levels of modularization, including monolithic,

strategic, tactical and a combination of both. The results showed that the combination of both levels obtained the better results (see Figure 3).

On additional experiments, tactical modularity was implemented for an Aibo robot. In this case, 31 tactical modules were required to generate the controller. The controller was generated to solve different tasks like stand up, standing and pushing the ground (Téllez et al., 2005). The controller was also able to generate one of the first MNNs controller able to make Aibo walk (Téllez et al., 2006).

FUTURE TRENDS

Within the evolutionary robotics paradigm, it is very difficult to generate complex behaviours when the robot used is quite complex with a huge number of sensors and actuators. The use of tactical modularity together with strategic one, is introduced as a possible solution to the problem of generating complex behaviours in complex robots. Even if some examples have been

Figure 3. This figure represent the maximal performance value obtained by different types of modular approaches. Approach (a) is a monolithic approach, (b) and (c) are two different types of strategic approaches, (d) is tactical approach, and (f) is a reduced version of the tactical approach.
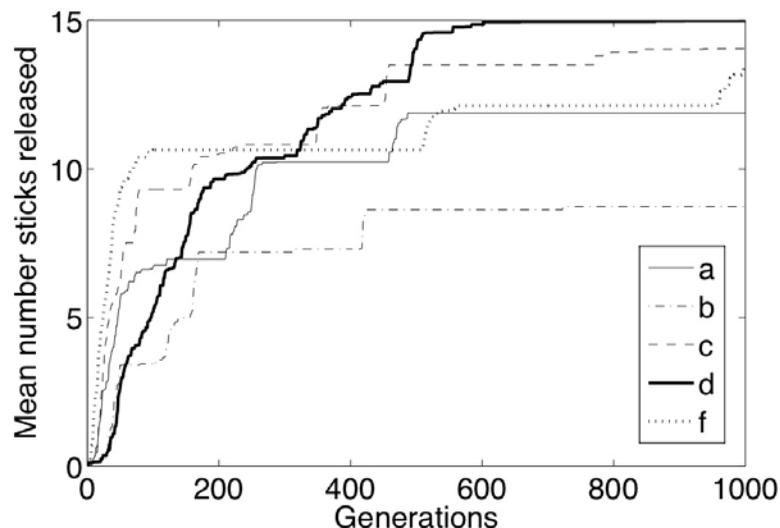
provided with a quite complex robot, it is necessary to see if the system can scale to systems with hundreds of elements.

Additional applications include its use in more classical domains like pattern recognition, speech recognition.

CONCLUSION

The level of modularity in neural controllers can be highly increased if tactical modularity is taken into account. This type of modularity complements typical modularization approaches based in strategic modularizations, by dividing strategic modules into their minimal components, and assigning one single neural module to each of them. This modularization allows the implementation of decomposable problems within a modularized structure. Both types of modularizations can be combined in order to obtain a highly modular neural controller, which shows better results in complex robot control.

REFERENCES

Auda, G., & Kamel, M. (1999), Modular neural networks: a survey, International Journal of Neural Systems, 9(2), 129-151.

Azam, F. (2000), Biologically inspired modular neural networks, PhD Thesis at the Virginia Polytechnic Institute and State University.

Boers, E., & Kuiper, H. (1992), Biological metaphors and the design of modular artificial neural networks, Master Thesis, Leiden University.

Caelli, G. L., & Wen, W. (1999), Modularity in neural computing, Proceedings of the IEEE 87(9), 1497-1518.

Fodor, J. (1983), The modularity of mind, The MIT Press.

Callebaut, W. (2005), The ubiquity of modularity, Modularity. Understanding the Development and Evolution of Natural Complex Systems, The MIT Press.

De Jong, E.D., & Thierens, D., & Watson, R.A. (2004), Defining Modularity, Hierarchy, and Repetition, Proceedings of the GECCO Workshop on Modularity,

regularity and hierarchy in open-ended evolutionary computation.

Di Ferdinando, A., & Calabretta, R., & Parisi, D. (2000), Evolving modular architectures for neural networks, Proceedings of the sixth Neural Computation and Psychology Workshop: Evolution, Learning and Development.

Garibay 0., & Garibay I., & Wu A.S. (2004), No Free Lunch for Module Encapsulation, Proceedings of the Modularity, Regularity and Hierarchy in Openended Evolutionary Computation Workshop - GECCO 2004.

Jacobs, R.A. (1990), Task decomposition through competition in a modular connectionist architecture, PhD thesis, University of Massachusets.

Jacobs, R.A., & Jordan, M.I., & Barto, A.G. (1991), Task decomposition through competition in a modular connectionist architecture: the what and where vision tasks, Cognitive Science, 15, 219-250.

Simon, H.A., (1969) The sciences of the artificial, The MIT Press.

Téllez, R.A., & Angulo, C., & Pardo, D. (2005), Highly modular architecture for the general control of autonomous robots, Proceedings of the 8th International Work-Conference on Artificial Neural Networks.

Téllez, R.A., & Angulo, C., & Pardo, D. (2006), Evolving the walking behaviour of a 12 DOF quadruped using a distributed neural architecture, Proceedings of the 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology.

Téllez, R., & Angulo, C. (2006) Tactical modularity for evolutionary animats, Proceedings of the International Catalan Conference on Artificial Intelligence.

Téllez, R., & Angulo, C. (2007), Acquisition of meaning through distributed robot control, Proceedings of the ICRA Workshop on Semantic information in robotics.

Watson, R.A., & Hornby, G.S., & Pollack, J. (1998), Modeling Building-Block Interdependency, Late Breaking Papers at the Genetic Programming 1998 Conference.

Watson, R. (2002), Modular Interdependency in Complex Dynamical Systems, Proceedings of the 8th Inter-

national Conference on the Simulation and Synthesis of Living Systems.

KEY TERMS

Cost Function: A mathematical function used to determine how good or how bad has a neural network performed during the training phase. The cost function usually indicates what is expected from the neural controller.

Element: Any variable of the program that contains a value that is used to feed into the neural network controller (input element) or to contain the answers of the neural network (output element). The input elements are usually the variables that contain the information from which the output will be generated. The output elements contain the output of the neural controller.

Evolutionary Robotics: A technique for the creation of neural controllers for autonomous robots, based on genetic algorithms.

Genetic Algorithm: An algorithm that simulates the natural evolutionary process, applied the generation of the solution of a problem. It is usually used to obtain the value of parameters difficult to calculate by other means (like for example the neural network weights). It requires the definition of a cost function.

Modularization: A process to determine the simplest meaningful parts that compose a task. There is no formal process to implement modularization, and in practice, it is very arbitrary.

Neural Controller: It is a computer program, based on artificial neural networks. The neural controller is a neural net or group of them which act upon a series of meaningful inputs, and generates one or several outputs.

M

Morphological Filtering Principles

Jose Crespo
Universidad Politécnica de Madrid, Spain

INTRODUCTION

In the last fifty years, approximately, advances in computers and the availability of images in digital form have made it possible to process and to analyze them in automatic (or semi-automatic) ways. Alongside with general signal processing, the discipline of image processing has acquired a great importance for practical applications as well as for theoretical investigations. Some general image processing references are (Castleman, 1979) (Rosenfeld & Kak, 1982) (Jain, 1989) (Pratt, 1991) (Haralick & Shapiro, 1992) (Russ, 2002) (Gonzalez & Woods, 2006).

Mathematical Morphology, which was founded by Serra and Matheron in the 1960s, has distinguished itself from other types of image processing in the sense that, among other aspects, has focused on the importance of shapes. The principles of Mathematical Morphology can be found in numerous references such as (Serra, 1982) (Serra, 1988) (Giardina & Dougherty, 1988) (Schmitt & Mattioli, 1993) (Maragos & Schafer, 1990) (Heijmans, 1994) (Soille, 2003) (Dougherty & Lotufo, 2003) (Ronse, 2005).

BACKGROUND

Morphological processing especially uses set-based approaches, and it is not frequency-based. This is in fact in sharp contrast with linear signal processing (Oppenheim, Schafer, & Buck, 1999), which deals mainly with the frequency content of an input signal. Let us mention also that Mathematical Morphology (as the name suggests) normally employs a mathematical formalism.

Morphological filtering is a type of image filtering that focuses on increasing transformations. Shapes can be satisfactorily processed by morphological filters. Starting with elementary transformations that are based on Minkowski set operations, other more complex transformations can be realized. The theory of morphological filtering is soundly based on mathematics.

This article provides an overview of morphological filtering. The main families of morphological filters are discussed, taking into consideration the possibility of computing hierarchical image simplifications. Both the binary (or set) and gray-level function frameworks are considered.

In the following of this section, some fundamental notions of morphological processing are discussed. The underlying algebraic structure and associated operations, which establish the distinguishing characteristics of morphological processing, are commented.

UNDERLYING ALGEBRAIC STRUCTURE AND BASIC OPERATIONS

In morphological processing, the underlying algebraic structure is a complete *lattice* (Serra, 1988). A complete lattice is a set of elements with a partial ordering relationship, which will be denoted as $\leq$, and with two operations defined called *supremum* (sup) and *infimum* (inf):

- The sup operation computes the smallest element that is larger than or equal to the operands. Thus, if a, b are two elements of a lattice, "a sup b" is the element of the lattice that is larger than both a and b, and there is no smaller element that is so.
- The inf operation computes the greatest element that is smaller than or equal to the operands.

Moreover, every subset of a lattice has an infimum element and a supremum element.

For sets and gray-level images, these operations are:

- Sets (or binary images)
  - Order relationship: $\subseteq$ (set inclusion).
  - "A sup B" is equal to "A $\cup$ B", where A and B are sets.
  - "A inf B" is equal to "A $\cap$ B".

- Gray-level images (images with intensity values within a range of integers)
  - o Order relationship: For two functions *f*,*g*:

    $$f \leq g \Rightarrow f(x) \leq g(x),$$

    for all pixel *x*
    where the right-hand-side $\leq$ refers to the order relationship of integers.
  - o The sup of *f* and *g* is the function:

    $$(f \sup g)(x) = \max \{f(x), g(x)\}$$

    where "max" denotes the computation of the maximum of integers.
  - o The inf of *f* and *g* is the function:

    $$(f \inf g)(x) = \min \{f(x), g(x)\}$$

    where "min" symbolizes the computation of the minimum of integers.

TRANSFORMATION PROPERTIES

The concept of ordering is key in non-linear morphological processing, which focuses especially on those transformations that preserve ordering. An *increasing* transformation Ψ defined on a lattice satisfies that, for all *a*,*b*:

$$a \leq b \Rightarrow \Psi(a) \leq \Psi(b)$$

The following two properties concern the ordering between the input and the output. If *I* denotes an input image, an image operator Ψ is *extensive* if and only if, $\forall I$,

$$I \leq \Psi(I)$$

A related property is the anti-extensivity property. An operator Ψ is *anti-extensive* if and only if, $\forall I$,

$$I \geq \Psi(I)$$

The concept of *idempotence* is a fundamental notion in morphological image processing. An operator Ψ is idempotent if and only if, $\forall I$,

$$\Psi(I) = \Psi \Psi(I)$$

Within the non-linear morphological framework, the important *duality* principle states that, for each morphological operator, there exists a dual one with respect to the complementation operation.

Two operators Ψ and Ω are dual if

$$\Psi = C\Omega C$$

The complementation operation C, for sets, computes the complement of the input. In the case of gray-level images a related operation is the image inversion, which inverts an image reversing the intensity values with respect to the middle point of the intensity value range.

The following concept of *pyramid* applies to multi-scale transformations. A family of operators $\{\Psi_i\}$, where $i \in S = \{1,...,n\}$, forms a multi-level pyramid if

$$\forall j, k \in S, j \geq k, \exists l \text{ such that } \Psi_j = \Psi_l \Psi_k$$

In words, the set of transformations $\{\Psi_i\}$ constitutes a pyramid if any level *j* of the hierarchy can be reached by applying a member of $\{\Psi_i\}$ to a finer (smaller index) level *k*.

STRUCTURING ELEMENTS

A *structuring element* is a basic tool used by morphological operators to explore and to process the shapes and forms that are present in an input image. Normally, *flat* structuring elements, which are sets that define a shape, are employed. Two usual shapes (square and diamond) are displayed next (the "x" symbol denotes the center):

(a) Square 3x3   *(b) Diamond 3x3*
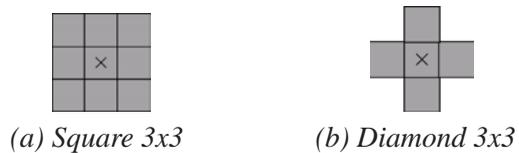
If *B* denotes a structuring element, its transposed is $\breve{B} = \{(-x,-y) \in B\}$ (i.e., *B* inverted with respect to the coordinate origin). If a structuring element *B* is centered and symmetric, then $\breve{B} = B$.

DILATIONS AND EROSIONS

Dilations and *erosions* are the most basic transformations in morphological processing. Dilations δ are increasing operators that satisfy, $\forall I, I'$,

$$\delta(I \sup I') = \delta(I) \sup \delta(I')$$

Respectively, erosions ε are increasing operators that satisfy, $\forall I, I'$,

$$\varepsilon (I \inf I') = \varepsilon(I) \inf \varepsilon(I')$$

Dilations and erosions by structuring element perform, respectively, sup and inf operations over an input image that depend on a structuring element B. These dilations and erosions are symbolized, respectively, by δ_B and ε_B, and they originate from the Minkowski set addition and subtraction. Let us first discuss the set framework.

In the set framework, if A denotes an input set, the $\delta_B(A)$ dilation computes the locus of points where the B structuring element translated to has a non-empty intersection with (i.e., "touches") input set A:

$$\delta_B(A) = \{x \mid B_x \cap A \neq \varnothing\}$$

B_x symbolizes the structuring element B translated to point (or pixel) x (i.e., $B_x = \{x' \mid x' - x \in B\}$, where "-" symbolizes the vector subtraction).

Figure 1 shows a set example in R^2. Input set A (composed of two connected-components) and structuring element B (a circle) are displayed in part (a). The $\delta_B(A)$ dilation is shown in part (b).

The previous expression can be formulated using the sup operation as:

$$\delta_B(A) = \bigcup_{b \in B} A_{-b} = \sup_{b \in B} A_{-b}$$

Using the lattice framework, the expression for functions is formally identical to the expression for sets:

$$\delta_B(I) = \sup_{b \in B} I_{-b}$$

Note: the sup operator is that of the function lattice.

The function expression can be written in another way that gives a more operational expression to compute the value of the result of $\delta_B(I)$ at each pixel x of I:

$$[\delta_B(I)](x) = \max_{b \in B}\{I(x + b)\}$$

Figure 1. Dilation $\delta_B(A)$

(a) Set A and structuring element B

(b) $\delta_B(A)$
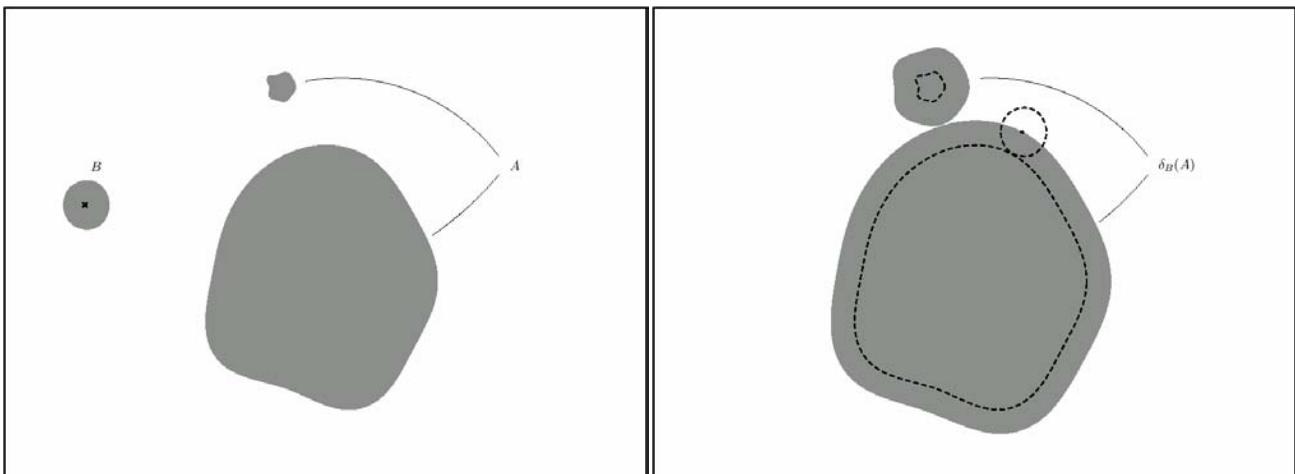
The sup operation has been replaced by the "max" operation that computes the maximum of a set of integers. Note that "$[\delta_B(I)](x)$" is the intensity value of pixel x in that image. The "+" symbol denotes the vector addition.

Some important properties of dilations δ_B are the following:

- For sets, dilation δ_B is commutative, i.e., if A denotes an input set, then $\delta_B(A) = \delta_A(B)$.
- If a structuring element B contains the coordinate origin, then δ_B is extensive.
- The dilation by a structuring element is associative, i.e, if B is the result of $\delta_C(D)$ (or $\delta_D(C)$), then $\delta_B(I) = \delta_C(\delta_D(I)) = \delta_D(\delta_C(I))$.

If A denotes an input set and B denotes a structuring element, the $\varepsilon_B(A)$ erosion computes the locus of points where the B structuring element translated to is completely included within input set A:

$$\varepsilon_B(A) = \{x \mid B_x \subseteq A\}$$

Figure 2 displays a set example of $\varepsilon_B(A)$, where A is that of the previous dilation example.

The following expressions of erosions ε_B are analogous to those already introduced for dilations.

Figure 2. Erosion $\varepsilon_B(A)$

$$\varepsilon_B(A)$$

The expression for sets formulated by means of the inf operation is:

$$\varepsilon_B(A) = \bigcap_{b \in B} A_{-b} = \inf_{b \in B} A_{-b}$$

The expressions for functions are:

$$\varepsilon_B(I) = \inf_{b \in B} I_{-b}$$

$$[\varepsilon_B(I)](x) = \min_{b \in B}\{I(x + b)\}$$

Some important properties of erosions ε_B are the following:

- If the coordinate origin belongs to a structuring element B, then ε_B is anti-extensive.
- The erosion by a structuring element is associative, i.e, if B is the result of $\delta_C(D)$ (or $\delta_D(C)$), then $\varepsilon_B(I) = \varepsilon_C(\varepsilon_D(I)) = \varepsilon_D(\varepsilon_C(I))$.

In fact, expressions for erosions are dual of, respectively, those of dilations; δ_B and ε_B are dual of each other:

$$\delta_B = C\, \varepsilon_B\, C$$

A simple 1-D example of a gray-level dilation and erosion (where B has 3 points) is the following:

9	10	11	12	12	11	13	13	12	12	12	10	9	9	9	10

$$I$$

10	11	12	12	12	13	13	13	13	12	12	12	10	9	10	10

$$\delta_B(I)$$

9	9	10	11	11	11	11	12	12	12	10	9	9	9	9	9

$$\varepsilon_B(I)$$
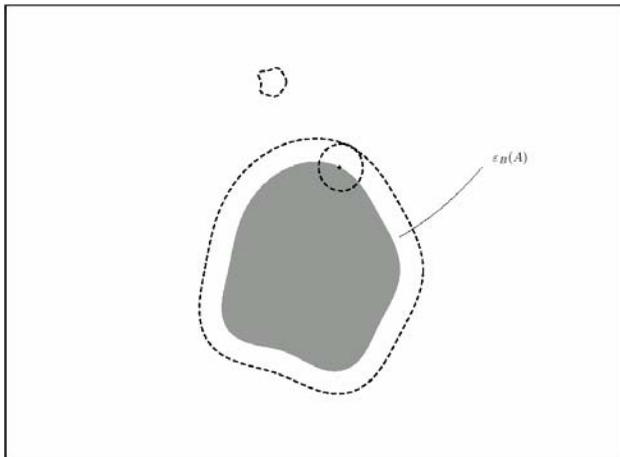
FROM SET OPERATORS TO FUNCTION OPERATORS

Flat operators are function operators Ψ that can be derived from a set operator Ψ' that satisfy the following threshold superposition property:

$$[\Psi(I)](x) = \sup\{u{:}x \in \Psi'\,(U_u(I))\}$$

where U_u is the thresholding operator at level u, and I is an image. The thresholding operator at level u is defined as

$$U_u(I) = \{x : I(x) \geq u\}$$

Let us define a variant of the thresholding operator that outputs a binary function (instead of a set): $(U'_u(I))(x)$ is 1 if $I(x) \geq u$, and 0 otherwise. Then, a flat operator Ψ that commutes with thresholding is said to satisfy:

$$U'_u \Psi = \Psi U'_u$$

BASIC MORPHOLOGICAL FILTERS

Openings and Closings

In morphological processing, a *filter* is an increasing and idempotent transformation.

The two most fundamental filters in morphological processing arise when there is an order between the input and the filter output. They are the so-called *openings* and *closings*, symbolized, respectively, by γ and φ.

- An *opening* γ is an anti-extensive morphological filter.
- A *closing* φ is an extensive morphological filter.

The names "algebraic openings" and "algebraic closings" are also used in the literature to refer to these most general types of openings and closings.

The computation of openings and closings that use structuring elements as the "shape probes" to process input image shapes is discussed next. They are defined in terms of dilations and erosions by structuring element.

For an input set A, an opening by structuring element B, symbolized by γ_B, is the set of points x that belong to a translated structuring element that *fits* a set A, i.e., that is included in A. Let us establish how γ_B is computed in the next definition, which applies both to sets and images.

An *opening by structuring element B*, symbolized by γ_B is defined by

$$\gamma_B = \delta_B \varepsilon_B$$

i.e., γ_B is the sequential composition of an erosion ε_B followed by a dilation $\delta_{\breve{B}}$, where $\breve{B}$ denotes B transposed.

This type of filter first erodes an input image by the ε_B erosion, and then the subsequent $\delta_{\breve{B}}$ dilation generally *recovers* in some sense the parts of the input image that have persisted. Nevertheless, not everything is normally recovered, and the output image is always less than or equal to the input image.

The definition of the dual closing follows. A *closing by structuring element B*, symbolized by φ_B is defined by

$$\varphi_B = \varepsilon_{\breve{B}} \delta_B$$

i.e., φ_B is the sequential composition of a dilation δ_B followed by an erosion $\varepsilon_{\breve{B}}$, where $\breve{B}$ denotes B transposed.

Alternated Filters

The sequential compositions of an opening γ and a closing φ are called *alternated* sequential compositions.

A morphological *alternated filter* is a sequential composition of an opening and a closing, i.e.,

$$\varphi\gamma \text{ and } \gamma\varphi,$$

are alternated filters.

An important fact is that there is generally no ordering between the input and output of alternated filters, i.e.,

$$I \not\leq \varphi\gamma\,(I) \not\leq I$$

$$I \not\leq \gamma\varphi\,(I) \not\leq I$$

In addition, there is generally no ordering between $\varphi\gamma$ and $\gamma\varphi$.

Alternated filters are quite useful in image processing and analysis because they combine in some way the effects of both openings and closings in one filter.

Parallel Combination Properties

The class of openings (or, respectively, of closings) is closed under the sup (respectively, inf) operation. In other words:

- The sup of openings is an opening.
- The inf of closings is a closing.

Different structuring elements can be combined to achieve a desired shape filtering effect. For example, the effect of a sup of openings such as (γ_A sup γ_B), which is itself an opening, can be quite different from either γ_A or γ_B.

GRANULOMETRIES AND ANTI-GRANULOMETRIES

M

The *granulometry* concept formalizes the *size distribution* notion. Size distributions are families of transformations Ψ_i with a size parameter i that satisfy the following axioms:

Figure 3. Granulometry
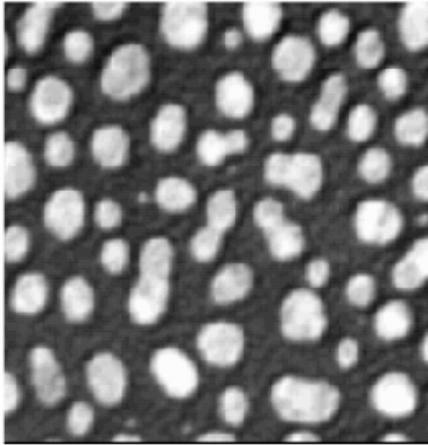
(a) Input image I
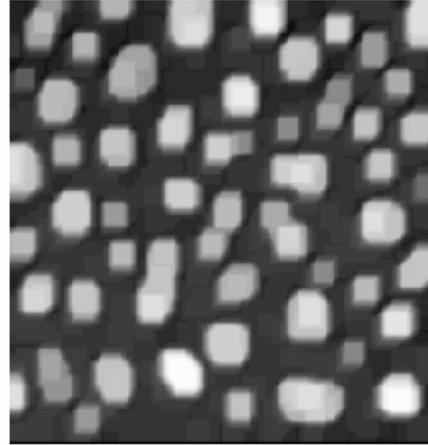
(b) $\gamma_4(I)$
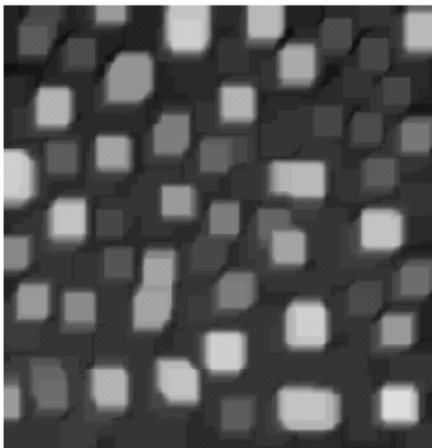
(c) $\gamma_6(I)$
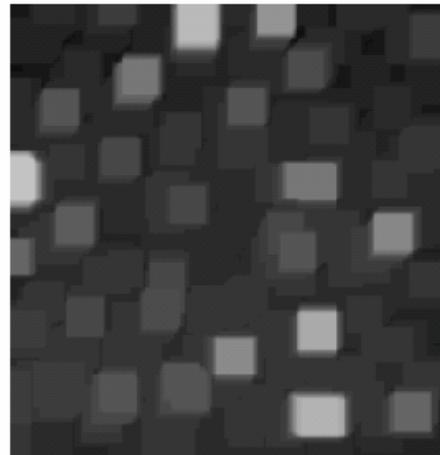
(d) $\gamma_8(I)$

- *Increasingness*
- *Anti-extensivity*
- *Absorption*

If Ψ_i, Ψ_j belong to a size distribution, where $i \leq j$, then

$$\Psi_i \Psi_j = \Psi_j \Psi_i = \Psi_{\max(i,j)}$$

In morphological filtering, the so-called *granulometries* are families of transformations that satisfy the size distribution axioms above. A family of openings $\{\gamma_i\}$, where $i \in S = \{1,...,n\}$ is a granulometry if, for all $i, j \in S$,

$$i \leq j \Rightarrow \gamma_i \geq \gamma_j.$$

i.e., an ordered family of openings constitutes a granulometry.

The dual concept of a granulometry is called an *anti-granulometry*, which is an ordered family of closings, as defined next. A family of closings $\{\varphi_i\}$, where $i \in S = \{1,...,n\}$ is an anti-granulometry if, for all $i, j \in S$,

$$i \leq j \Rightarrow \varphi_i \leq \varphi_j.$$

Quite often, both a granulometry and an anti-granulometry are computed.

Normally, quantitative increasing measures are computed at each output. These measure values build a curve that can be used to characterize an input image if the measure criterion is appropriate.

$$M(\gamma_N(I) \leq ... \geq M\ (\gamma_1(I) \leq M(I) \leq M(\varphi_1(I) \leq ... \geq M (\varphi_N(I)$$

An example of an increasing criterion is the area or number of pixels in binary images, or the volume in non-binary images.

To build a family of openings and a family of closings by structuring elements that constitute, respectively, a granulometry and antigranulometry, an appropriate family of structuring elements $\{iB \mid i \in \{0,...,N\}\}$ that ensure the ordering of openings and closings is needed. Particularly, the family of structuring elements must satisfy the following property:

$$\gamma_{(i-1)B}(iB) = iB, \text{ for } i \geq 1.$$

Figure 3 illustrates the granulometry concept. Three opening outputs have been displayed in parts (b), (c) and (d), particularly the outputs corresponding to openings γ_4, γ_6 and γ_8, where the subindex indicates the size of the structuring element (Iñesta & Crespo, 2003). A subindex *i* refers to a square of side 2i+1. There is an ordering between the four images: part (a) $\geq$ part (b) $\geq$ part (c) $\geq$ part (d).

MULTI-LEVEL MORPHOLOGICAL FILTERING

Alternating Sequential Filters

Granulometries and anti-granulometries allow to build complex filters composed of ordered openings and closings.

An *alternating sequential filter* ASF is an ordered sequential composition of alternated filters $\varphi_j\gamma_j$ or $\gamma_j\varphi_j$, such as

$$ASF_i = \varphi_i\gamma_i...\varphi_j\gamma_j...\varphi_1\gamma_1$$

$$ASF'_i = \gamma_i\varphi_i...\gamma_j\varphi_j...\gamma_1\varphi_1$$

where $i \geq j \geq 1$, and where γ_i and φ_i belong, respectively, to a granulometry and an anti-granulometry.

Alternating sequential filters satisfy the following absorption property: if $i \geq j$, then

$$ASF_i \, ASF_j = ASF_i$$

$$ASF_j \, ASF_i \leq ASF_i$$

$$ASF'_i \, ASF'_j = ASF'_i$$

$$ASF'_j \, ASF'_i \geq ASF'_i$$

Morphological Pyramids

Multi-level (or multi-scale) operators are families of transformations that depend on a scale parameter *i*. Within morphological filters, cases that satisfy the pyramid condition are:

- granulometries,
- anti-granulometries, and
- alternating sequential filters.

FUTURE TRENDS

Operators that consider connectivity aspects have been an active research and work area in morphological processing. Connectivity integrates easily in the morphological filtering framework using the connected class concept (and the associated opening) introduced in (Serra, 1988).

The class of *connected filters* (Serra & Salembier, 1993) (Crespo, Serra, & Schafer, 1993) (Vincent, 1993) (Salembier & Serra, 1995) (Crespo, Serra, & Schafer, 1995) (Breen & Jones, 1996) (Crespo & Schafer, 1997) (Crespo & Maojo, 1998) (Garrido, Salembier, & Garcia, 1998) (Heijmans, 1999) (Crespo, Maojo, Sanandrés, Billhardt, & Muñoz, 2002) (Crespo & Maojo, 2008), which preserve shapes particularly well, has been successfully used in image processing and analysis applications. In more recent years, certain types of connected filters, such as the so-called levelings (Meyer, 1998) (Meyer, 2004), whose origin in the set framework can be traced back to (Crespo et al., 1993) (Crespo & Schafer, 1997), have been the focus of new research efforts.

CONCLUSION

This article has provided a summary of morphological filtering, which is qualitatively different from linear filtering. These differences are clear when morphological filtering is approached analysing the underlying algebraic framework and the key importance of ordering and increasingness.

Morphological filtering provides a distinctive type of image analysis that is appropriate to deal with shapes. Although in its origin morphological filtering was especially associated to set processing (and many concepts are originally set-based), it extends to non-binary gray-level functions.

ACKNOWLEDGMENTS

REFERENCES

Breen, E. J., & Jones, R. (1996, November). Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding*, *64*(3), 377-389.

Castleman, K. (1979). *Digital image processing*. Englewood Cliffs: Prentice Hall.

Crespo, J., & Maojo, V. (1998, April). New results on the theory of morphological filters by reconstruction. *Pattern Recognition*, *31*(4), 419-429.

Crespo, J., Maojo, V. (2008). *The strong property of morphological connected alternated filters.* Accepted for publication in the Journal of Mathematical Imaging and Vision. DOI: 10.1007/x10851-008-0098-x.

Crespo, J., Maojo, V., Sanandrés, J., Billhardt, H., & Muñoz, A. (2002). On the strong property of connected open-close and close-open filters. In J. Braquelaire, J.-O. Lachaud, & A. Vialard (Eds.), *Discrete geometry for computer imagery* (Vol. 2301, pp. 165-174). Berlin-Heidelberg: Springer-Verlag.

Crespo, J., & Schafer, R.W. (1997). Locality and adjacency stability constraints for morphological connected operators. *Journal of Mathematical Imaging and Vision*, *7*(1), 85–102.

Crespo, J., Serra, J., & Schafer, R. W. (1993, May). Image segmentation using connected filters. In J. Serra & P. Salembier (Eds.), *Workshop on mathematical morphology, Barcelona* (pp. 52–57).

Crespo, J., Serra, J., & Schafer, R. W. (1995, November). Theoretical aspects of morphological filters by reconstruction. *Signal Processing*, *47*(2), 201–225.

Dougherty, E., & Lotufo, R. (2003). *Hands-on morphological image processing*. Bellingham: SPIE Press.

Garrido, L., Salembier, P., & Garcia, D. (1998). Extensive operators in partition analysis for image sequence analysis. *Signal Processing*, *66*(2), 157–180.

Giardina, C., & Dougherty, E. (1988). *Morphological methods in image and signal processing*. Englewood Clliffs: Prentice-Hall.

Gonzalez, R. C., & Woods, R. E. (2006). *Digital image processing* (3rd. ed.). Englewoods Cliff: Prentice Hall.

M

Haralick, R., & Shapiro, L. (1992). *Computer and robot vision. Volume I*. Reading: Addison-Wesley Publishing Company.

Heijmans, H. (1994). *Morphological image operators*. Boston: Academic Press.

Heijmans, H. (1999). Connected morphological operators for binary images. *Computer Vision and Image Understanding*, *73*, 99–120.

Iñesta, J. M., & Crespo, J. (2003). Principios básicos del análisis de imágenes médicas. In M. Belmonte, O. Coltell, V. Maojo, J. Mateu, & F. Sanz (Eds.), *Manual de informática médica* (pp. 299–333). Barcelona: Editorial Menarini - Caduceo Multimedia.

Jain, A. (1989). *Fundamentals of digital image processing (Prentice hall information and system sciences series; Series Editor: T. Kailath)*. Englewood Cliffs: Prentice Hall.

Maragos, P., & Schafer, R. W. (1990, April). Morphological systems for multidimensional signal processing. *Proc. of the IEEE*, *78*(4), 690–710.

Meyer, F. (1998). From connected operators to levelings. In H. J. A. M. Heijmans & J. B. T. M. Roerdink (Eds.), *Mathematical morphology and its applications to image and signal processing* (pp. 191–198). Dordrecht: Kluwer Academic Publishers.

Meyer, F. (2004, January-March). Levelings, image simplification filters for segmentation. *Journal of Mathematical Imaging and Vision*, *20*(1-2), 59–72.

Oppenheim, A., Schafer, R. W., & Buck, J. (1999). *Discrete-time signal processing* (2nd. ed.). Englewood Cliffs: Prentice-Hall.

Pratt, W. (1991). *Digital image processing* (2nd. ed.). New York: John Wiley and Sons.

Ronse, C. (2005). Guest editorial. *Journal of Mathematical Imaging and Vision*, *22*(2 - 3), 103–105.

Rosenfeld, A., & Kak, A. (1982). *Digital picture processing - Volumes 1 and 2*. Orlando: Academic Press.

Russ, J. C. (2002). *The image processing handbook* (4th. ed.). Boca Raton: CRC Press.

Salembier, P., & Serra, J. (1995). Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing*, *4*(8), 1153–1160.

Schmitt, M., & Mattioli, J. (1993). *Morphologie mathématique*. Paris: Masson.

Serra, J. (1982). *Mathematical morphology. Volume I*. London: Academic Press.

Serra, J. (Ed.). (1988). *Mathematical morphology. Volume II: Theoretical Advances*. London: Academic Press.

Serra, J., & Salembier, P. (1993, July). Connected operators and pyramids. In *Proceedings of SPIE, Nonlinear algebra and morphological image processing, San Diego* (Vol. 2030, pp. 65–76).

Soille, P. (2003). *Morphological image analysis* (2nd. ed.). Berlin-Heidelberg-New York: Springer-Verlag.

Vincent, L. (1993, April). Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, *2*, 176–201.

KEY TERMS

Duality: The duality principle states that, for each morphological operator, there exists a dual one. In sets, the duality is established with respect to the set complementation operation (see further details in the text).

Extensivity: A transformation is extensive when its output is larger than or equal to the input. Anti-extensivity is the opposite concept: a transformation is anti-extensive when its output is smaller than or equal to the input.

Idempotence: A transformation Ψ is said to be idempotent if, when sequentially applied twice, it does not change the output of the first application, i.e., $\Psi \Psi = \Psi$.

Image Transformation: An operation that processes an input image and produces an output image.

M

Increasingness: A transformation is increasing when it preserves ordering. If Ψ is increasing, then $a \leq b \Rightarrow \Psi(a) \leq \Psi(b)$**.**

Lattice: A complete lattice is a set of elements with a partial ordering relationship and two operations called *supremum* and *infimum*.

Morphological Filter: An increasing and idempotent transformation.

Multi-Scale Transformation: A transformation that displays some characteristics controllable by means of (at least) a parameter, which is called the size or scale parameter.

MREM, Discrete Recurrent Network for Optimization

Enrique Mérida-Casermeiro
University of Málaga, Spain

Domingo López-Rodríguez
University of Málaga, Spain

Juan M. Ortiz-de-Lazcano-Lobato
University of Málaga, Spain

INTRODUCTION

Since McCulloch and Pitts' seminal work (McCulloch & Pitts, 1943), several models of discrete neural networks have been proposed, many of them presenting the ability of assigning a discrete value (other than unipolar or bipolar) to the output of a single neuron. These models have focused on a wide variety of applications. One of the most important models was developed by J. Hopfield in (Hopfield, 1982), which has been successfully applied in fields such as pattern and image recognition and reconstruction (Sun et al., 1995), design of analog digital circuits (Tank & Hopfield, 1986), and, above all, in combinatorial optimization (Hopfield & Tank, 1985) (Takefuji, 1992) (Takefuji & Wang, 1996), among others.

The purpose of this work is to review some applications of multivalued neural models to combinatorial optimization problems, focusing specifically on the neural model MREM, since it includes many of the multivalued models in the specialized literature.

BACKGROUND

In Hopfield and Tank's pioneering work (Hopfield & Tank, 1985), neural networks were applied for the first time to solve combinatorial optimization problems, concretely the well-known travelling salesman problem. They developed two types of networks, discrete and continuous, although the latter has been mostly chosen to solve optimization problems, adducing that it helps to escape more easily from local optima. Since then, the search for better neural algorithms, to face the diverse problems of combinatorial optimization (many of them

belonging to the class of NPcomplete problems), has been the objective of researchers in this field.

This method of optimization consists of minimizing an energy function, whose parameters and constraints are obtained by means of identification with the objective function of the optimization problem. In this case, the energy function has the form:

$$E(\boldsymbol{S}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{i,j} s_i s_j + \sum_{i=1}^{N} \theta_i s_i$$

where N is the number of neurons of the network, $w_{i,j}$ is the synaptic weight between neurons j and i, and θ_i is the threshold or bias of the neuron i.

In the discrete version of Hopfield's model, component s_i of the state vector $\boldsymbol{S} = (s_1,...,s_N)$ can take values in $\mathcal{M} = \{-1,1\}$ (constituting the bipolar model) or in $\mathcal{M} = \{0,1\}$ (unipolar model). In the continuous version, $\mathcal{M} = [-1,1]$ or $\mathcal{M} = [0,1]$. This continuous version, although it has been traditionally the most used for optimization problems, presents certain inconveniences:

- Certain special mechanisms, maybe in form of constraints, should be contributed in order to get that, in the final state of the network, all the components of state vector $\boldsymbol{S}$ belong to $\{-1, 1\}$ or $\{0,1\}$.
- The traditional dynamics used in this model, implemented in a digital computer, does not guarantee the decrease of the energy function in every iteration, so it is not ensured that the final state is a minimum of the energy function (Galán-Marín, 2000).

However, the biggest problem of this model (the discrete as well as the continuous one) is the possibility to converge to a non feasible state, or to a local (not global) minimum. Wilson and Pawley (1988) demonstrated, through massive simulations, that, for the travelling salesman problem of 10 cities, only 8% of the solutions were feasible, and most not good. Moreover, this proportion got worse when problem size was increased.

After this, many works were focused on improving Hopfield's network:

- By modifying the energy function (Xu & Tsai, 1991).
- By adjusting the numerous parameters present in the network, as in (Lai & Coghill, 1988).
- By using stochastic techniques in the dynamics of the network (Kirkpatrick et al., 1983) (Aarts & Korst, 1988).

Particularly, researchers tried to improve the efficiency of Hopfield's network for the travelling salesman problem, achieving acceptable results, but inferior to Operations Research techniques (Takahashi, 1997). The reason for these disappointing results is that the linear formulation used by these techniques is a great advantage in comparison with neural networks, which unavoidably use a quadratic energy function, impeding the use of subpaths deletion techniques (Smith, 1996), and provoking the appearance of a bigger number of local minima.

Another research line was devoted to the improvement of Hopfieldtype recurrent networks, and their application to diverse problems of optimization, in which some results proved to be better than those obtained by traditional Operations Research techniques (Smith & Krishnamoorthy, 1998). Takefuji's work (Takefuji, 1992) (Lee et al., 1992)(Takefuji & Wang, 1996), with a great number of publications in international media, must be highlighted. Their results have been overcome by the OCHOM model (GalánMarín & MuñozPérez, 2001).

MULTIVALUED DISCRETE RECURRENT MODEL. APPLICATION TO COMBINATORIAL OPTIMIZATION PROBLEMS

M

A new generalization of Hopfield's model arises in the works (MéridaCasermeiro, 2000) (MéridaCasermeiro et al., 2001), where the MREM (Multivalued REcurrent Model) model is presented.

The Neural MREM Model

This model presents two essential features that make it very versatile and that increase its applicability:

- The output of each neuron, s_i, is a value of the set $\mathcal{M} = \{m_1, m_2, \ldots, m_L\}$, which is not necessarily numeric.
- The concept of similarity function f between neuron outputs is introduced. $f(x,y)$ represents the similarity between neuron states x and y.

This way, the energy function of this model is as follows:

$$E(\boldsymbol{S}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{i,j} f(s_i, s_j) + \sum_{i=1}^{N}\theta_i(s_i)$$

where $\theta_i : \mathcal{M} \rightarrow \mathbb{R}$ is a generalization of the thresholds of each neuron.

The features mentioned above make that in this model certain optimization problems (as the travelling salesman problem), have a better representation than in the unipolar or bipolar Hopfield's models, and their successors.

It is clear that MREM includes Hopfield's models (with outputs in $\mathcal{M} = \{-1,1\}$ or in $\mathcal{M} = \{0,1\}$) if we consider the similarity function given by the product $f(a,b) = ab$. Other multivalued models, like MAREN or SOAR (Erdem & Ozturk, 1996) (Ozturk & Abut, 1997), are also generalized by MREM.

The dynamics for this network is chosen according to the problem to be tackled.

Application to Several Combinatorial Optimization Problems

This multivalued model has been successfully applied to diverse optimization problems, outperforming the best-established algorithms. Several of these applications can be found at (MéridaCasermeiro et al., 2003) (MéridaCasermeiro & LópezRodríguez, 2005) (López-Rodríguez et al., 2006).

These problems are typical representatives of the NPcomplete complexity class, indicating their degree of difficulty in resolution.

The Travelling Salesman Problem

Traveling Salesman Problem (TSP) is one of the most wellknown and studied combinatorial optimization problems due to its wide range of reallife applications and intrinsic complexity.

Reallife applications cover aspects such as automatic routing for robots and hole location in printed circuits design (Reinelt, 1994), as well as gas turbine checking, machine task scheduling or crystallographic analysis (Bland & Shallcross, 1987), among others.

This problem can be stated as follows: given N cities $X_1,...,X_N$ and distances $d_{i,j}$ between each pair of cities X_i and X_j, the objective is to find the shortest closed tour visiting each city once.

In order to get the TSP solved by this neural model, two identifications must be done:

- A network state must be identified to a solution to the TSP: Since a solution to the N cities TSP can be represented as a permutation in the set of numbers $\{1,...,N\}$, the net will be formed by N neurons, taking value in the set $\mathcal{M} = \{1,\ldots,N\}$, such that state vector $S = (s_1,...,s_N)$ represents a permutation of $\{1,...,N\}$. With this representation, $s_i = k$ means that kth city will be visited in the ith place.

- The energy function must be identified to the total distance of a tour: If we let $f(x,y) = -2d_{x,y}$ and

$$w_{i,j} = \begin{cases} 1, & (j = i+1) \vee ((i = N) \wedge (j = 1)) \\ 0, & \text{otherwise} \end{cases}$$

the energy function obtained is

$$E(S) = \sum_{i=1}^{N-1} d_{s_i,s_{i+1}} + d_{s_N,s_1},$$

the total distance of the tour represented by state vector S.

Computational dynamics is based on starting with a random feasible initial state vector and updating neuron outputs to keep the current state vector inside the feasible states set. To this end, at each iteration, a 2opt update will be made on current state vector, that is, every pair of neurons, p,q with $p > q + 1$, is studied and checked in parallel whether there exists a cross between segments (s_p, s_{p+1}) and (s_q, s_{q+1}). In this case, the next relation holds:

$$d_{s_p,s_{p+1}} + d_{s_q,s_{q+1}} > d_{s_p,s_q} + d_{s_{p+1},s_{q+1}}$$

Then, the trajectory between cities s_{p+1} and s_q is inverted, that is, if S is the current state, the new state vector S' will be defined by:

$$s_i' = \begin{cases} s_{q+p+1-i}, & p+1 \le i \le q \\ s_i, & \text{otherwise} \end{cases}$$

As an additional technique for improvement, it has also been considered 3opt updates: the tour is decomposed into three consecutive arcs, A, B and C, which are then recombined in all possible ways: {ABC,ACB,AB'C,ABC',AB'C',AC'B, ACB', AC'B'}, where A',B',C' are the reversed arcs corresponding to A, B, and C, respectively. Note that {ABC,AB'C,ABC',AC'B'} are 2opt updates, so there is no need to check them again.

The next state of the net will be the combination that decreases most the energy function. Further details in (MéridaCasermeiro et al., 2003).

In (MéridaCasermeiro et al., 2003), some experimental results are provided, for problems from the TSPLIB repository (see Table 1). This model is compared against KNIES (Aras et al., 1999), a model based on Kohonen's self organizing map. MREM proved to outperform KNIES, obtaining in many cases almost optimal solutions.

Figure 1. Best solution found by MREM (left, error=1.3%) and optimal solution (right)
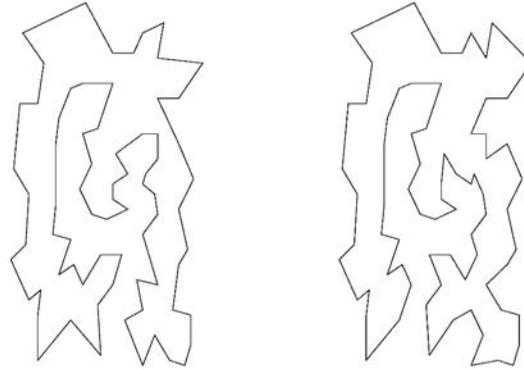
Table 1. Results of KNIES and MREM for the TSP for some instances from TSPLIB

Instance	Optimum	KNIES	MREM		
		Best (%)	Best (%)	Av. (%)	t (sec)
eil51	426	2.86	0.23	2.43	3.12
st70	675	1.51	0.00	1.89	9.01
eil76	538	4.98	1.30	3.43	10.80
rd100	7910	2.09	0.00	3.02	61.70
eil101	629	4.66	1.43	3.51	27.76
lin105	14379	1.29	0.00	1.71	28.83
pr107	44303	0.42	0.15	0.82	49.79
pr124	59030	0.08	0.00	1.23	59.51
bier127	118282	2.76	0.42	2.06	66.29
kroA200	28568	5.71	3.49	6.70	318.44

The Graph Partition Problem

Let $\mathcal{G} = (\mathcal{V}, \varepsilon)$ be an undirected graph without self-connections. $\mathcal{V} = \{v_i\}$ is the set of vertices and ε is the set of n_e edges. For each edge $(v_i, v_j) \in \varepsilon$ there is a weight $c_{i,j} \in \mathbb{R}^+$. All weights can be expressed by a symmetric real matrix C, with $c_{i,j} = 0$ when it does not exist the arc (v_i, v_j).

MaxCut Problem: to find a partition of $\mathcal{V}$ into K disjoint sets A_i such that the sum of the weights of the

edges from ε, that have their endpoints in different elements of the partition, is maximum. Therefore, the function to maximize is

$$\sum_{i|v_i \in A_m} \sum_{\substack{j|v_j \in A_n \\ m>n}} c_{i,j}$$

To solve the MaxCut problem with MREM, we need N neurons, one per node in $\mathcal{V}$. The output of neuron i, $s_i \in \mathcal{M} = \{1, 2, \ldots, K\}$, will denote that ith node is assigned to A_{s_i}.

Since it is equivalent to maximize the cost of edges cut by the partition and to minimize the cost of edges with endpoints in the same set of the partition, the objective function can be modelled as an energy function by taking $w_{i,j} = -2c_{i,j}$ and $f(x,y) = \delta_{x,y}$ (that is, $f(x,y) = 1$ if, and only if, $x = y$, otherwise it is 0), considering $\theta_i = 0$.

The dynamics used in (MéridaCasermeiro & López-Rodríguez, 2005) was named best2.

best2 consists in getting the greatest decrease of the energy function by changing the state of only two neurons at each time. If neurons p and q are to be updated, energy increments $\Delta E(i,j)$ when $s_p = i$ and $s_q = j$, for $i, j \in \{1,\ldots,K\}$, are computed. Then, the state of minimum increase is chosen as the new network state.

By using this dynamics, in (MéridaCasermeiro & LópezRodríguez, 2005), the MREM model is compared against some other networks, like OCHOM (Galán-Marín & MuñozPérez, 2001), obtaining the best results in authors' experiments (see Table 2).

Table 2. Results for MaxCut comparing MREM and OCHOM

N	dens	MREM			OCHOM		
		Best	Av.	t	Best	Av.	t
	0,05	276,8	256,28	0,05	276,8	242,15	0,0023
	0,25	1013,2	970,84	0,06	999,6	926,26	0,0026
50	0,5	1778,8	1724,08	0,06	1778,8	1694,44	0,0033
	0,75	2663,6	2475,48	0,05	2646	2432,47	0,0036
	0,9	2941,8	2876,18	0,06	2940,4	2865,83	0,0031
	0,05	990,2	917,72	0,15	958,8	867,64	0,0064
	0,25	3719,2	3620,9	0,14	3725,5	3571,24	0,0086
100	0,5	6711,6	6637,08	0,13	6695,8	6585,54	0,0126
	0,75	9816,2	9524,1	0,14	9816,2	9444,33	0,0118
	0,9	11348,8	11215,06	0,14	11391,3	11148,4	0,0109
	0,05	2009,8	1933,6	0,26	1929,6	1837,43	0,0147
	0,25	7990	7807,16	0,26	7940,2	7690,35	0,0258
150	0,5	14701,4	14531,06	0,24	14658,4	14489,5	0,0209
	0,75	21126,2	20899,94	0,22	21124	20907,6	0,0252
	0,9	24926	24589,62	0,22	24859,7	24533,1	0,0256
	0,05	3411,4	3321,84	0,38	3409,5	3316,28	0,0276
	0,25	13741	13533,9	0,35	13617,9	13439,7	0,0468
200	0,5	25750,8	25500,18	0,34	25770,8	25526,8	0,0451
	0,75	37038,6	36789,2	0,32	36932	36683,4	0,0486
	0,9	43584,8	43296,26	0,33	43420,6	43104,6	0,0462

The 2Pages Graph Layout Problem

In the last years, several graph representation problems have been studied in the literature. Most of them are related to the linear graph layout problem, in which the vertices of a graph are placed along a horizontal ``node line'', or ``spine'' (dividing the plane into two halfplanes or ``pages'') and then edges are added to this representation as specified by the adjacency matrix. The objective of this problem is to minimize the number of crossings produced by such a layout.

Some examples of problems associated to this linear graph layout problem (or 2 pages crossing number problem, 2PCNP) are the bandwidth problem (Chinn et al., 1982), the book thickness problem (Kainen, 1990), the pagenumber problem (Malitz, 1994), the boundary VLSI layout problem (Ullman, 1984) and the singlerow routing problem (Raghavan & Sahni, 1983), or printed circuit board layout (Sinden, 1966) and automated graph drawing (Tamassia et al., 1988).

In (LópezRodríguez et al., 2007), a neural model, derived from MREM, is designed to solve this problem. One of the differences of this model with the algorithms developed in literature is that there is no need of assigning a good ordering of the vertices at a preprocessing step. The model, as well as the relative position of the arcs, computes this optimal node order.

To solve the 2PCNP problem, authors have considered two MREM neural models:

*   The first network will be formed by N neurons, being N the number of nodes in the graph. Neurons output (the state vector) indicate the node ordering in the line. Thus, $s_i = k$ will be interpreted as the kth node being placed in the ith position in the node line. Hence, the output of each neuron can take value in the set $\mathcal{M}_1 = \{1, 2, \ldots, N\}$.
*   The second network will be formed by as many neurons as edges in the graph, M. The output of each neuron will belong to the set $\mathcal{M}_2 = \{-1, 1\}$. For the arc (v_i, v_j), $S_{(v_i, v_j)} = -1$ will indicate that the edge will be drawn in the lower halfplane, and $S_{(v_i, v_j)} = +1$, in the upper one.

Initially, the state of the net of vertices is randomly selected as a permutation of $\{1, 2, \ldots, N\}$. At any time,

Table 3. Comparison between MREM and the heuristics mentioned in Cimikowski's work

Graph	N	M	MREM	CN	e-len	1-page	greedy
K_6	6	21	3	3	3	4	5
K_7	7	28	9	9	11	9	13
K_8	8	36	18	18	18	30	27
K_9	9	45	36	36	42	50	50
K_{10}	10	55	60	60	80	92	80
$C_{20}(1, 2)$	20	40	0	2	0	0	0
$C_{20}(1, 2, 3)$	20	60	19	24	36	48	40
$C_{20}(1, 2, 3, 4)$	20	80	74	74	90	118	108
$C_{22}(1, 2, 3)$	22	66	22	26	40	54	44
$C_{22}(1, 3, 5, 7)$	22	88	198	200	306	294	286
$C_{24}(1, 3)$	24	48	11	14	22	16	22
$C_{26}(1, 3)$	26	52	11	16	24	16	24
$C_{28}(1, 3, 5)$	28	84	80	86	138	138	130
$C_{30}(1, 3, 5)$	30	90	92	96	148	150	140

Figure 2. Optimal layouts for graphs K$_6$ (left) and K$_{3,3}$ (right)
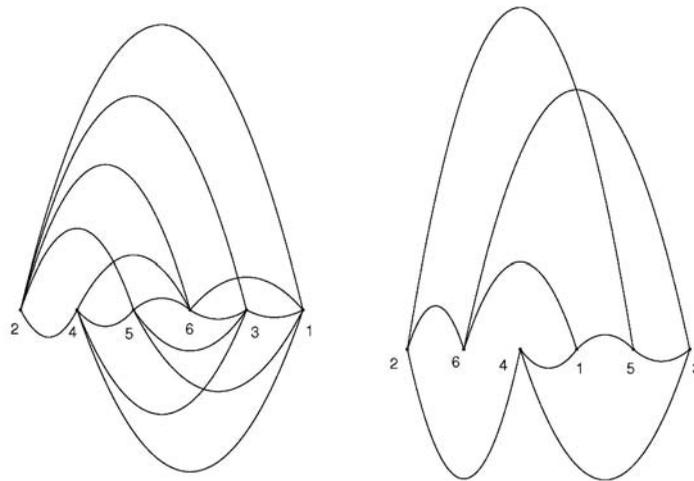
the net is looking for a better solution than the current one, in terms of minimizing the energy function. This is achieved by permuting the output of two neurons (node positions) and changing the location of an edge (from the upper halfplane to the lower one, and viceversa).

In (LópezRodríguez et al., 2007), this new model is compared against some heuristics (Cimikowski, 2002) specially designed for this problem. MREM obtained the best solutions in the experiments, improving the best known solution in some cases (Table 3).

FUTURE TRENDS

Recurrent neural networks can be used to solve many optimization problems. Researchers and practitioners can benefit from the application of the neural model MREM to diverse optimization problems.

Other problems where these models can be applied cover aspects such as data classification, image compression by vector quantization, etc. It must be noted that many graph-based problems can be easily formulated in terms of minimizing the energy function of this model: degreeconstrained minimum spanning tree, maximum clique, etc.

CONCLUSION

The first works in optimization by neural networks were inspired in Hopfield's models. These models did not obtain good results when compared to the wellknown Operations Research techniques.

Many researchers focused on developing new neural models to improve the performance of Hopfieldtype networks in this kind of tasks.

The problem of these binary models is that all the information given by the problem has to be specified by means of only two values ($\{0,1\}$ or $\{-1,1\}$), so some information is lost.

Multivalued neural models are designed to represent the information of the problem by means of more than two values, achieving a better representation of the problem.

With this improvement, computational dynamics of multivalued models can be easily designed to solve a given optimization problem. These advantages make this kind of networks a very powerful ally in tackling combinatorial problems.

The MREM model is a multivalued model that generalizes many other models, so it can be easily used to solve optimization problems, as shown in the text.

Some applications of the model are wellknown NPcomplete optimization problems, like the Traveling Salesman Problem, the Graph Partition Problem, and the 2 Pages Crossing Number Problem. As shown in the references, this model is able to outperform the bestalgorithmuptodate in each of the mentioned problems.

REFERENCES

Aarts, E. & Korst, J. (1988). Simulated Annealing with Boltzman Machines. Wiley.

Aras, N., Oomen B.J., & Altinel, I.K. (1999). The Kohonen Network Incorporating Explicit Statistics and its Application to the Travelling Salesman Problem. Neural Networks, 12:1273-1284.

Bland, R. & Shallcross, D. F. (1987). Large Traveling Salesman Problem Arising from Experiments in Xray Crystallography: a Preliminary Report on Computation. Technical Report No. 730, School of OR/IE, Cornell University, New York.

Chinn, P. Z., Chvátalová, L., Dewdney, A. K., & Gibbs, N. E. (1982). The bandwidth problem for graphs and matrices-a survey. J. Graph Theory, 6, 223-253.

Cimikowski, R. (2002). Algorithms for the fixed linear crossing number problem. Discrete Applied Mathematics, 122, 93 – 115.

Erdem, M. H. & Ozturk, Y. (1996). A new family of multivalued networks. Neural Networks, 9(6), 979-989.

GalánMarín, G. & MuñozPérez, J. (2001). Design and analysis of maximum hopfield networks. IEEE Trans. on Neural Networks, 12(2), 329-339.

GalánMarín, G. (2000). Redes Neuronales Recurrentes para Optimización Combinatoria. PhD thesis, Universidad de Málaga.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities, 79, 2254-2558.

Hopfield, J. & Tank, D. (1985). Neural computation of decisions in optimization problems. Biological Cybernetics, 52, 141-152.

Kainen, P. C. (1990). The book thickness of a graph, ii. Congr. Numer., 71, 127-132.

Kirkpatrick, S., Gellat, C. D., & Vecchi, M. (1983). Optimization by simulated annealing. Science, 220, 671-680.

Lai, W. K. & Coghill, G. G. (1988). Genetic breeding of control parameters for the hopfieldtank neural network. In Proc. Int. Conf. Neural Networks (pp. 618-623).

Lee, K. C., Funabiki, N., & Takefuji, Y. (1992). A parallel improvement algorithm for the bipartite subgraph problem. IEEE Transactions on Neural Networks, 3(1), 139-145.

LópezRodríguez, D., MéridaCasermeiro, E., Ortiz de LazcanoLobato, J. M., & GalánMarín, G. (2007). Two pages graph layout via recurrent multivalued neural networks. Lecture Notes in Computer Science, 4507, 192-199.

LópezRodríguez, D., MéridaCasermeiro, E., Ortiz de LazcanoLobato, J. M., & LópezRubio, E. (2006). Image compression by vector quantization with recurrent discrete networks. Lecture Notes in Computer Science, 4132, 595-605.

Malitz, S. M. (1994). On the page number of graphs. J. Algorithms, 17(1), 71-84.

McCulloch, W. & Pitts, W. (1943). A logical calculus of the ideas inmanent in nervous activity. Bulletin Mathematical Biophysic, 5, 115-133.

MéridaCasermeiro, E. (2000). Red Neuronal recurrente multivaluada para el reconocimiento de patrones y la optimización combinatoria. PhD thesis, Universidad de Málaga.

MéridaCasermeiro, E., GalánMarín, G., & Muñoz Pérez, J. (2001). An efficient multivalued Hopfield network for the travelling salesman problem. Neural Processing Letters, 14, 203-216.

MéridaCasermeiro, E. & López Rodríguez, D. (2005). Graph partitioning via recurrent multivalued neural networks. Lecture Notes in Computer Science, 3512, 1149-1156.

MéridaCasermeiro, E., Muñoz Pérez, J., & Domínguez-Merino, E. (2003). An nparallel multivalued network: Applications to the travelling salesman problem.

Computational Methods in Neural Modelling, Lecture Notes in Computer Science, 2686, 406-413.

Ozturk, Y. & Abut, H. (1997). System of associative relationships (SOAR). Proceedings of ASILOMAR.

Raghavan, R. & Sahni, S. (1983). Single row routing. IEEE Trans. Comput, C32(3), 209-220.

Reinelt, G. (1994). The Travelling Salesman. Computational Solutions for TSP Applications. Springer.

Sinden, F. W. (1966). Topology of thin films circuit. Bell Syst. Tech. Jour., XLV, 1639-1666.

Smith, K. (1996). An argument for abandoning the traveling salesman problem as a neural network benchmark. IEEE Transactions on Neural Networks, 7(6), 1542-1544.

Smith, K., P. M. & Krishnamoorthy, M. (1998). Neural techniques for combinatorial optimization with aplications. IEEE Transactions on Neural Networks, 9(6), 1301-1318.

Sun, Y., Li, J. G., & Yu, S. Y. (1995). Improvement on performance of modified Hopfield network for image restoration. IEEE Transactions on Image Processing, 5, 688-692.

Takahashi, Y. (1997). Mathematical improvement of the Hopfield model for TSP feasible solutions by synapse dynamical systems. Neurocomputing, 15, 15-43.

Takefuji, Y. (1992). Neural Network Parallel Computing. Kluwer Academic.

Takefuji, Y. & Wang, J. (1996). Neural Computing for Optimization and Combinatorics, volume 3. World Scientific.

Tamassia, R., Di Battista, G., & Batini, C. (1988). Automatic graph drawing and readability of diagrams. IEEE Trans. Syst. Man. Cybern., SMC18, 61-79.

Tank, D. & Hopfield, J. (1986). Simple neural optimization networks: An a/d converter, signal decision circuit and linear programming circuit. IEEE Transactions on Circuits and Systems, 33(5), 533-541.

Ullman, J. D. (1984). Computational Aspects of VLSI. Computer Science Press.

Wilson, V. & Pawley, G. (1988). On the stability of the TSP problem algorithm of Hopfield and Tank. Biological Cybernetics, 58, 63-70.

Xu, X. & Tsai, W. T. (1991). Effective neural algorithms for the traveling salesman problem. Neural Networks, 4, 193-205.

KEY TERMS

2 Pages Graph Layout Problem: Problem of finding an ordering of the nodes of a graph on a straight line, and assigning, to each edge, a location in any of the two halfplanes induced by that line, such that the number of crossings between edges is minimum.

Artificial Neural Network: Structure for distributed and parallel processing of information, formed by a series of units (which may possess a local memory and make local information processing operations), interconnected via one-way communication channels, called connections.

Computational Dynamics: Updating scheme of the neuron outputs in a neural model.

Energy Function: Objective function of the optimization problem solved by a neural model.

MaxCut Problem: Problem of finding a partition of the set of nodes of a weighted graph, such that the sum of the costs corresponding to edges, with end-points in different sets of the partition, is maximum.

Multivalued Discrete Neural Model: A model of neural networks in which neuron outputs may take value in the set $\mathcal{M} = \{m_1, \ldots, m_L\}$, instead of $\mathcal{M} = \{-1, 1\}$ or $\mathcal{M} = \{0, 1\}$.

Travelling Salesman Problem: Problem of finding the shortest closed tour that visits a series of N cities. Each city must be visited exactly one time.

Multilayer Optimization Approach for Fuzzy Systems

Ivan N. Silva
University of São Paulo, Brazil

Rogerio A. Flauzino
University of São Paulo, Brazil

INTRODUCTION

The design of fuzzy inference systems comes along with several decisions taken by the designers since is necessary to determine, in a coherent way, the number of membership functions for the inputs and outputs, and also the specification of the fuzzy rules set of the system, besides defining the strategies of rules aggregation and defuzzification of output sets. The need to develop systematic procedures to assist the designers has been wide because the trial and error technique is the unique often available (Figueiredo & Gomide, 1997).

In general terms, for applications involving system identification and fuzzy modeling, it is convenient to use energy functions that express the error between the desired results and those provided by the fuzzy system. An example is the use of the mean squared error or normalized mean squared error as energy functions. In the context of systems identification, besides the mean squared error, data regularization indicators can be added to the energy function in order to improve the system response in presence of noises (from training data) (Guillaume, 2001).

In the absence of a tuning set, such as happens in parameters adjustment of a process controller, the energy function can be defined by functions that consider the desired requirements of a particular design (Wan, Hirasawa, Hu & Murata, 2001), i.e., maximum overshoot signal, setting time, rise time, undamped natural frequency, etc.

From this point of view, this article presents a new methodology based on error backpropagation for the adjustment of fuzzy inference systems, which can be then designed as a three layers model. Each one of these layers represents the tasks performed by the fuzzy inference system such as fuzzification, fuzzy rules inference and defuzzification. The adjustment procedure proposed in this article is performed through the adaptation of its free parameters, from each one of these layers, in order to minimize the energy function previously specified.

In principle, the adjustment can be made layer by layer separately. The operational differences associated with each layer, where the parameters adjustment of a layer does not influence the performance of other, allow single adjustment of each layer. Thus, the routine of fuzzy inference system tuning acquires a larger flexibility when compared to the training process used in artificial neural networks. This methodology is interesting, not only for the results presented and obtained through computer simulations, but also for its generality concerning to the kind of fuzzy inference system used. Therefore, such methodology is expandable either to the Mandani architecture or also to that suggested by Takagi-Sugeno.

BACKGROUND

In the last years it has been observed a wide and crescent interest in applications involving logic fuzzy. These applications include from consumer products, such as cameras, video camcorders, washing machines and microwave ovens, even industrial applications as control of processes, medical instrumentation and decision support systems (Ramot, Friedman, Langholz & Kandel, 2003).

The fuzzy inference systems can be treated as methods that use the concepts and operations defined by the fuzzy set theory and by fuzzy reasoning methods (Sugeno & Yasukawa, 1993). Basically, these operational functions include fuzzification of inputs, application of inference rules, aggregation of rules and defuzzification, which represents the crisp outputs of the fuzzy

system (Jang, 1993). At present time, there are several researchers engaged in studies related to the design techniques involving fuzzy inference systems.

The first type of design technique of fuzzy inference system has its focus addressed to enable the modeling of process from their expert knowledge bases, where both antecedent and consequent terms of the rules are always fuzzy sets, offering then a high semantic level and a good interpretability capacity (Mandani & Assilian, 1975). However, the applicability of this technique in the mapping of complex systems composed by several input and output variables has been an arduous task, which can produce as inaccurate results as poor performance (Guillaume, 2001)(Becker, 1991).

The second type of design technique of fuzzy inference system can be identified as being those that incorporate learning, in an automatic way, from data that are representing the behavior of the input and output variables of the process. Therefore, this design strategy uses a collection of input and output values obtained from the process to be modeled, which differs of the first design strategy, where the fuzzy system was defined using only the expert knowledge acquired from observation on the respective system. In a generic way, the methods derived from this second strategy can be interpreted as being composed by automatic generation techniques of fuzzy rules, which use the available data for their adjustment procedures (or training).

Among the main approaches belonging to this second design strategy, it has been highlighted the ANFIS (Adaptive-Network-based Fuzzy Inference Systems) algorithm proposed by Jang (1993), which is applicable to the fuzzy architectures constituted by real polynomial functions as consequent terms of the fuzzy rules, such as those presented by Takagi & Sugeno (1985) and Sugeno & Kang (1988). The more recent approaches, such as those proposed by Panella & Gallo (2005), Huang & Babri (2006) and Li & Hori (2006), are also belonging to this design strategy.

However, the representation of a process through these automatic architectures can implicate in interpretability reduction in relation to the created base of rules, whose consequent terms are expressed in most of the cases by polynomial functions, instead of linguistic variables (Kamimura, Takagi & Nakanishi,1994).

Thus, the development of adjustment algorithms of fuzzy inference systems, which the consequent terms of the fuzzy rules are also represented by fuzzy sets, has been widely motivated.

MAIN FOCUS OF THE CHAPTER

Considering the operational functions performed by the fuzzy inference systems, it is convenient to represent them by a three-layer model. Thus, the fuzzy inference

Figure 1. Fuzzy inference system composed by two inputs and one output
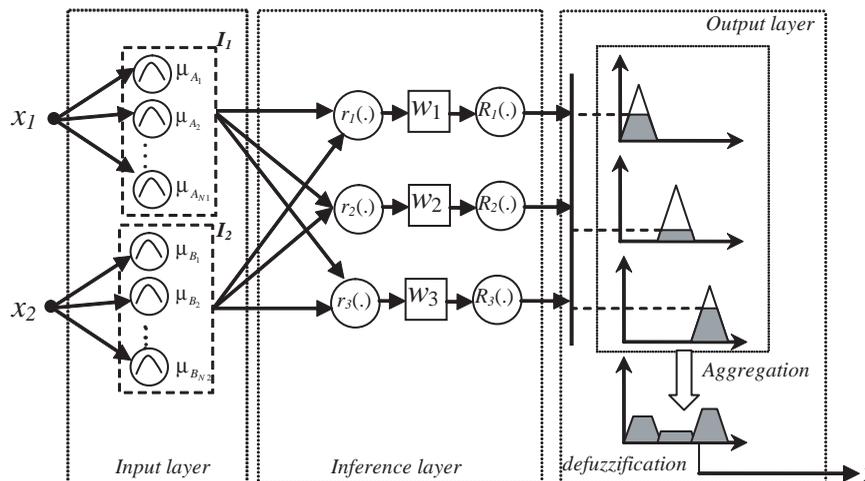
system presented in this article can be represented by the sequential composition of three layers, i.e., input layer, inference layer and output layer.

The input layer has functionalities of connecting the input variables (coming from outside) with the fuzzy inference system, performing their respective fuzzifications through proper membership functions. In the inference layer of the fuzzy rules, the input fuzzified variables are combined among them, according to defined rules, using as support the operations defined by the fuzzy theory. The resulting set of this aggregation process is then defuzzified to produce the fuzzy inference system output. The aggregation and defuzzification process of the fuzzy system output are both made by the output layer. It is important to observe, concerning to the output layer, that although it performs the two processes above described, it is also responsible for storing the membership functions of the output variables. As illustration, Fig. 1 shows the proposed multilayer model, which is constituted by two inputs and one output, having three fuzzy rules in its inference layer.

In the following subsections further details will be presented about how fuzzy inference systems can be represented by a three-layers model.

Input Layer

The inputs fuzzification has the purpose of determining the membership degree of each input related to the fuzzy sets associated with each input variable. To each input variable of the fuzzy system can be associated as many fuzzy sets as necessary. In this way, let a fuzzy system constituted by only one input with N fuzzy sets, the output of the input layer will be a column vector with N elements, which are representing the membership degrees of this input in relation to those fuzzy sets. If we define the input of this fuzzy system with a unique input x, then the output of the input layer will be the vector I_1 represented by:

$$I_1(x) = \begin{bmatrix} \mu_{A_1}(x) & \mu_{A_2}(x) & \cdots & \mu_{A_N}(x) \end{bmatrix}^T \tag{1}$$

where $\mu_{A_k}(.)$ is the membership function defined to input x, which is referring to the k-th fuzzy set associated with it.

The generalization of the input layer concept for a fuzzy system having p input variables can be achieved

if we consider each input being modeled as a sub-layer of the input layer. Taking into account this consideration, the output vector of the input layer $I(x)$ is then defined by:

$$I(x) = \begin{bmatrix} I_1(x_1)^T & I_2(x_2)^T & \cdots & I_p(x_p)^T \end{bmatrix}^T \tag{2}$$

where x_i is the i-th input of the fuzzy system and $I_k(.)$ is the k-th vector of membership functions associated with the input x_k. In Fig. 1 is illustrated the input layer for a fuzzy system composed by two inputs, which are mapped by the vectors I_1 and I_2.

There are several membership functions that can be used in the proposed approach. One of the necessary requisites for those functions is that they are normalized in the closed domain [0,1].

Inference Layer

The inference layer of a fuzzy system has the functionality of processing the fuzzy inference rules defined for it. Another functionality is to provide a knowledge base for the process.

In this paper, the fuzzy inference system has initially all the possible inferred rules. Therefore, the tuning algorithm has the task of weighting the inference rules. The weighting of the inference rules is a proper way to represent the most important rules, or even to allow that conflicting rules are related to each other without any linguistic completeness loss. Thus, it is possible to express the i-th fuzzy rule as follows:

$$R_i(I(x)) = w_i r_i(I(x)) \tag{3}$$

where $R_i(.)$ is the function representing the fuzzy weighting of the i-th fuzzy rule, w_i is the weight of the i-th fuzzy rule and $r_i(.)$ represents the fuzzy value of the i-th fuzzy rule. In Fig. 1, it is shown the composition involving $r_i(.)$ and $R_i(.)$ for the three fuzzy rules belonging to the inference layer.

Output Layer

The output layer of the fuzzy inference system aims to aggregate the inference rules as well as the defuzzification of the fuzzy set generated from the aggregation of these inference rules.

Besides the operational aspects, the aggregation and defuzzification methods must consider the requi-

sites of hardware performance in order to reduce the computational effort needed for processing the fuzzy system. In this paper, the output layer of the inference system is also adjusted. The adjustment of this layer occurs in a similar way to that occurred with the input layer of the fuzzy system. As example, an illustration representing the procedures involved with the output layer is also shown in Fig. 1.

Adjustment of the Fuzzy Inference System

Let a fuzzy system with two inputs, each one composed of three gaussian membership functions, with a total of five inference rules, and having an output defined by two gaussian membership functions. It is known that, for each gaussian membership function, two free parameters should be considered, i.e., the mean and the standard deviation. Consequently, the number of free parameters of the input layer is 12. For each inference rule, a weighting factor has been associated, resulting a total of 5 free parameters in the inference layer. In relation to the output layer, the same considerations used for the input layer are valid. Therefore, four free parameters are associated with the output layer.

Therefore, the mapping f between the input space x and the output space y may be defined by:

$$y = f\left(x, mf_{In}, w, mf_{Out}\right) \tag{4}$$

where mf_{In} is the parameter vector associated with the input membership functions, w is the weight vector of the inference rules, and mf_{Out} is the parameter vector associated with the output membership functions. Therefore, mf_{In}, w and mf_{Out} represent the free parameters of the fuzzy system, which can be rewritten as follows:

$$y = f\left(x, \Theta\right) \tag{5}$$

where Θ is the vector resulting from concatenation of the free parameters involved with the fuzzy system, i.e.

$$\Theta = \left[mf_{In}^{\,T}, w^T, mf_{Out}^{\,T}\right] \tag{6}$$

The energy function to be minimized, considering the fixed tuning set $\{x,d\}$, is defined by:

$$\xi \equiv \xi_{(x,y)}\!\left(\Theta\right) \tag{7}$$

where ξ represents the energy function associated with the fuzzy inference system f.

Unconstrained Optimization Techniques

Let an energy function $\xi_{(x,y)}(\Theta)$ differentiable in relation to free parameters of the fuzzy inference system. Thus, the objective is to find an optimum solution Θ^* subject to:

$$\xi(\Theta^*) \le \xi(\Theta) \tag{8}$$

Therefore, we can observe that to satisfy the condition expressed in (8), it is necessary to solve an unconstrained optimization problem to obtain the solution Θ^*, which is given by:

$$\Theta^* \equiv \arg\min_{\Theta} \xi\!\left(\Theta\right) \tag{9}$$

The condition that expresses the optimum solution in (9) can also be rewritten as follows:

$$\nabla\xi(\Theta^*) = 0$$

where ∇ is the gradient operator defined by:

$$\nabla\xi\!\left(\Theta\right) = \left[\frac{\partial\xi}{\partial\Theta_1}, \frac{\partial\xi}{\partial\Theta_2}, \cdots, \frac{\partial\xi}{\partial\Theta_m}\right]^T \tag{10}$$

There are several techniques used to solve unconstrained optimization problems. A detailed description of these methods can be found in Bertsekas (1999). The selection of the most proper method is related to the complexity associated with the energy function. For example, the Gauss-Newton method for unconstrained optimization can be more applicable in problems where the energy function is defined by:

$$\xi(\Theta) = \frac{1}{2}\sum_{i=1}^{m} e^2(i) \tag{11}$$

where $e(i)$ is the absolute error in relation to the i-th tuning pattern.

In this paper, a derivation of the Gauss-Newton method is used for tuning fuzzy inference system, which is defined by the expression following:

$$\Theta_{next} = \Theta_{now} - \frac{1}{2}\left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{g} \tag{12}$$

where $\mathbf{g}$ is the gradient of ξ expressed in (11) and $\mathbf{J}$ is the Jacobean matrix of e defined in (12). The optimization algorithm used was the Levenberg-Marquardt method (Marquardt, 1963), which can efficiently handle ill-conditioned matrices $\mathbf{J}^T\mathbf{J}$ by altering equation (13) as follows:

$$\Theta_{next} = \Theta_{now} - \frac{1}{2}\left(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}\right)^{-1}\mathbf{g} \tag{13}$$

The calculation of the matrices $\mathbf{J}$ and the vectors $\mathbf{g}$ were performed through the finite differences method.

Simulation Results

This section presents simulation results of the proposed methodology for the Mandani fuzzy model. In the two examples following the fuzzy system is used to model nonlinear functions. In the first example, a fuzzy inference system is used to predict the Mackey-Glass time series. In the second example, a two-dimensional *sinc* function is modeled by the fuzzy inference system.

Example 1: Modeling the Mackey-Grass Function

Using the adjustment methodology presented in this paper, a fuzzy inference system of Mandani type was developed with objective to predict the Mackey-Glass time series (Mackey & Glass, 1977), which is defined by:

$$\frac{dx(t)}{dt} = -b \cdot x(t) + \frac{a \cdot x(t - \tau)}{1 + x(t - \tau)^c} \tag{14}$$

where the values of the constants are usually assumed as $a = 0.2$, $b = 0.1$ and $c = 10$. The value for the delay constant τ was 17. The tuning set was constituted by 500 patterns. The input variables of the fuzzy inference system were four, which correspond to values $x(t - 18)$, $x(t - 12)$, $x(t - 6)$ and $x(t)$. As output variable was adopted $x(t + 6)$.

The fuzzy inference system was defined having 4 fuzzy sets attributed to each input variable and also to the output variable. A total of 64 inference rules have been used in the inference process.

The energy function of the system was defined as being the mean squared error between the desired values $x(t + 6)$ and the values $\bar{x}(t + 6)$, i.e.

$$\xi(\Theta) = \frac{1}{L}\sum_{i=1}^{L}\left[x_i(t + 6) - \bar{x}_i(t + 6)\right]^2 \tag{15}$$

where L is the number of data used in the tuning process (L=500).

After minimization of (16), the membership functions of the fuzzy inference system were adjusted as illustrated in Fig. 2.

In Fig. 3 is presented the prediction results provided by the fuzzy inference system for 1000 sample points.

The mean squared error of estimation for the proposed problem was 0.000598 with standard deviation of 0.02448. The prediction error for the 1000 sample points is shown in Fig. 4.

For comparison, it was developed a fuzzy inference system adjusted by the ANFIS (Adaptive Neural-Fuzzy Inference System). This fuzzy inference system was composed by 10 membership functions for each input, being the knowledge base constituted by 10 rules. The mean squared error of estimation for the proposed problem was 0.000165 with standard deviation of 0.0041.

Example 2: Modeling the Two-Input Sinc Function

In this example is used the proposed methodology to model a two-dimensional *sinc* function defined by:

$$z = sinc(x, y) = \frac{sin(x) \cdot sin(y)}{x \cdot y} \tag{16}$$

From uniformly distributed grid points into the input range [-10,10] x [-10,10] of (17), 225 tuning data pairs were obtained. The fuzzy inference system used here

Figure 2. Input membership functions
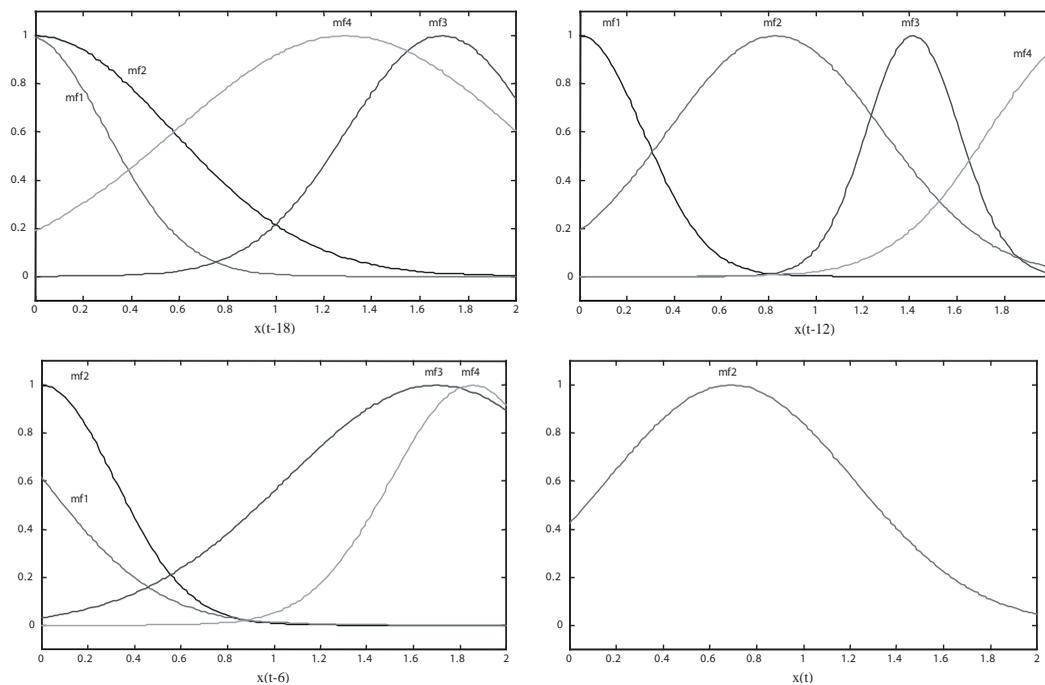
Fig. 3. Estimation of the fuzzy inference system for the Mackey-Glass series
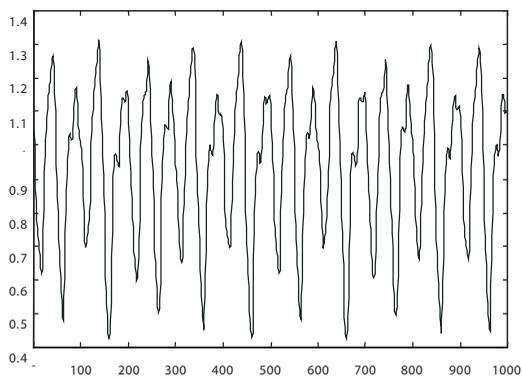
Figure 4. Prediction error for the Mackey-Glass series
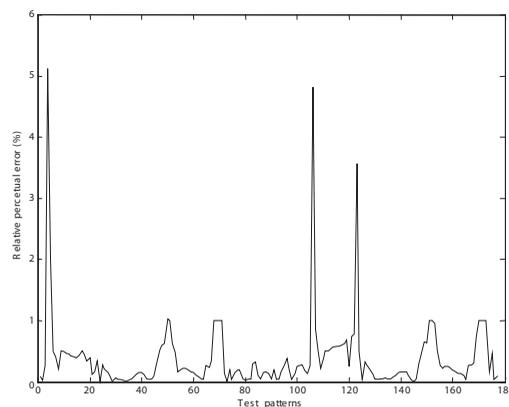
Figure 5. Tuning data (a) and reconstructed surface (b)

(a)   (b)
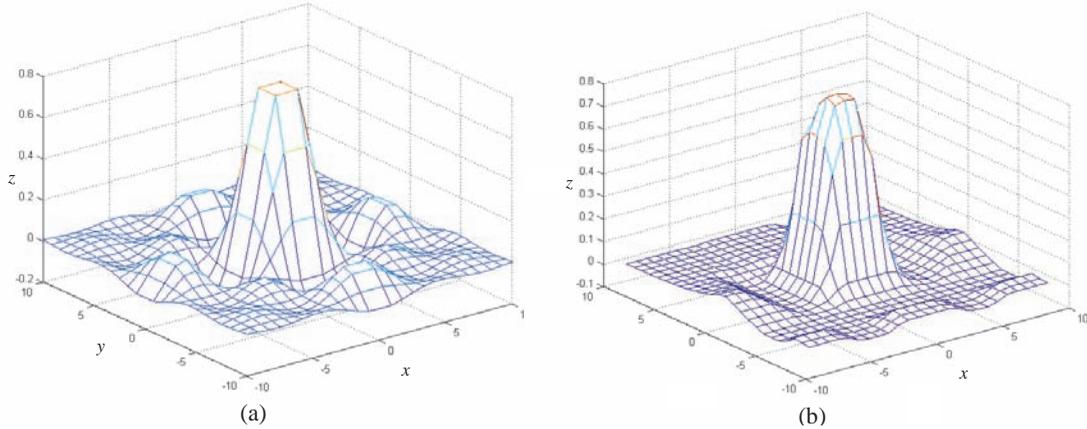
contains 11 rules, with 8 membership functions assigned to input variable x, 7 membership functions assigned to input y and 3 membership functions assigned to output z. The tuning data and the reconstructed surface are illustrated in Fig. 5.

FUTURE TRENDS

The methodology for the adjustment of fuzzy inference systems presented in this article can be considered very promising, not only for performance and precision obtained through computer simulations, but also for its interpretability in relation to the output variable, which is a highly desirable feature of a fuzzy system. Actually, it is the most prominent feature that distinguishes fuzzy systems from many other modeling techniques. We think that fuzzy system adjustment architectures, such as that proposed here, are ideally suited for explaining solutions to users because both premises (antecedents) and consequences of the rules are defined by fuzzy sets.

Future research and application should return to and concentrate on the linguistic features of fuzzy systems and their capabilities for knowledge representation, exploiting the tolerance for imprecision and uncertainty to summarize data and focus on decision-relevant information.

CONCLUSION

In this article was underlined the basic foundations involved with the adjustment process of fuzzy inference systems from unconstrained optimization techniques. In order to become the more efficient tuning, it is necessary that the energy function is properly specified for the adjustment process. To validation of the proposed methodology, the results obtained by the proposed approach were compared to those provided from the ANFIS methodology, and also through of mathematical modeling problems. The results obtained from this methodology offers new perspectives of researches related to the fuzzy inference systems, allowing thus that problems previously treated only by artificial neural networks may also be treated through fuzzy inference systems.

REFERENCES

Becker, S. (1991). Unsupervised Learning Procedures for Neural Networks. *International Journal of Neural Systems*. (2) 17-33.

Bertsekas, D. P. (1999). Nonlinear Programming (2nd ed.). Belmont: Athena Scientific.

Figueiredo, M., & Gomide, F. (1997). Adaptive Neuro Fuzzy Modeling. *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*. 1567-1572.

Guillaume, S. (2001). Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review. *IEEE Transactions on Fuzzy Systems*, (9) 426-443.

Huang, G.-B., & Babri, H.A. (2006). Universal Approximation Using Incremental Networks with Random Hidden Computation Nodes. *IEEE Transactions on Neural Networks*, (17) 4, 879-892.

Jang, J.R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics*. (23) 665-685.

Kamimura, R., Takagi, T., & Nakanishi, S. (1994). Improving Generalization Performance by Information Minimization. *Proceedings of the IEEE World Congress on Computational Intelligence*. 143-148.

Li, W., & Hori, Y. (2006). An Algorithm for Extracting Fuzzy Rules Based on RBF Neural Network. *IEEE Transactions on Industrial Electronics*, (53) 4, 1269-1276.

Mackey, M.C., & Glass, L. (1977). Oscillation and Chaos in Physiological Control Sciences. *Science*. (197) 287-289.

Mandani, E.H., & Assilian, S. (1975). An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*. (7) 1-13.

Marquardt, D. (1963). An Algorithm for Least Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics*. (11) 431-441.

Panella, M., & Gallo, A.S. (2005). An Input-output Clustering Approach to the Synthesis of ANFIS Networks. *IEEE Transactions on Fuzzy Systems*, (13) 1, 69-81.

Ramot, D., Friedman, M., Langholz, G., & Kandel, A. (2003). Complex Fuzzy Logic. *IEEE Transactions on Fuzzy Sets*. (11) 450-461.

Sugeno, M., & Kang, G.T. (1988). Structure Identification of Fuzzy Model. *Fuzzy Sets and Systems*. (28) 15-33.

Sugeno, M., & Yasukawa, T. (1993). A Fuzzy-logic-based Approach to Qualitative Modeling. *IEEE Transactions on Fuzzy Systems*. (1) 7-31.

Takagi, T., & Sugeno, M. (1985). Fuzzy Identification of System and Its Application to Modeling and Control. *IEEE Transactions on Systems, Man, and Cybernetics*. (15) 116-132.

Wan, W., Hirasawa, K., Hu, J., & Murata, J. (2001). Relation Between Weight Initialization of Neural Networks and Pruning Algorithms: Case Study on Mackey-Glass Time Series. *Proceedings of the International Joint Conference on Neural Networks*. 1750-1755.

KEY TERMS

Backpropagation Algorithm: Learning algorithm of ANNs, based on minimizing the error obtained from the comparison between the outputs that the network gives after the application of a set of network inputs and the outputs it should give (the desired outputs).

Defuzzification: Process of producing a quantifiable result (crisp) in fuzzy logic. Typically, a fuzzy system will have a number of rules that transform a number of variables into a "fuzzy" result, that is, the result is described in terms of membership in fuzzy sets.

Fuzzification: Process of transforming crisp values into grades of membership for linguistic terms of fuzzy sets. The membership function is used to associate a grade to each linguistic term.

Fuzzy Logic: Type of logic dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic.

Fuzzy Rule: Linguistic constructions of type IF-THEN that have the general form "IF *A* THEN *B*", where *A* and *B* are (collections of) propositions containing linguistic variables.

Fuzzy System: Approach of the computational intelligence that uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data.

Membership Function: Generalization of the indicator function in classical sets. In fuzzy logic, it represents the degree of truth as an extension of valuation.

M

Multi-Layered Semantic Data Models

László Kovács
University of Miskolc, Hungary

Tanja Sieber
University of Miskolc, Hungary

INTRODUCTION

One of the basic terms in information engineering is data. In our approach, data item is defined as representation of an information atom stored in digital computers. Although an information atom can be considered as a subject-predicate-value triplet (Lassila, 1999), data is usually given only with its value representation. This fact can lead to definitions where data is just numbers, words or pictures without context. For example in (WO, 2007), data is given as information in numerical form that can be digitally transmitted or processed. It is interesting that *we can often recognize that the term 'data' is used without any exact terminological definition with the effect that the term often remains confusing, sometimes even contradicting the definitions of the term presented. Sieber and Kammerer (2006) introduce a new interpretation of data containing several levels. The lowest level belongs to data instances that describe the form and appearance of symbols. The intermediate level is the level of representatives which includes the applied encoding system. The highest level is related to the meaning with context description. All three levels are needed* to get to know the information atom. For example the symbol '36' in a database determines only the value and representation system, but not the meaning. To cover the whole information atom, the database should store some additional data items to describe the original data. The main purpose of semantic data models is to describe both context and the main structure of data items in the problem area. These additional data items are called metadata. It is important to see that:

- metadata are data,
- metadata are relative, and
- metadata describe data.

Metadata constitute a basis for bringing together data that are related in terms of content, and for processing them further. They can be understood as a pre-requisite for intelligent and efficient administration and processing, and not least as a focused, formal means of providing relevant data.

BACKGROUND

In data management systems, the context of a value is usually defined with the help of a storage structure. An identification name (a text value) is assigned to each position of the structure. The description of storage (structure, naming and constraints) is called schema. A big problem of structural data modeling is that it can not provide all the information needed to understand the full context of the data. For example, a relational schema

RT (NM INT, KNEV CHAR(20), RU DATE)

alone is not enough to capture the meaning of the stored data items.

The main building blocks to describe the context in semantic data models (SDM) are concepts and relationships. The first widely known structure oriented semantic models in database design are the Entity-Relationship (ER) model (Chen, 1976) and the EER (Thalheim, 2000) model. The ER model consists of three basic elements: entity (concept), relationship and attribute. The attributes are considered as structure elements of the entities, one attribute may belong to only one entity. The EER model is the extension of the ER model with IS_A and HAS_A relationships. Some other extensions are SIM, IFO and RM/T. One of the main drawbacks of structure oriented SDM is the limitations of expressive power.

Later, models like UML or ODL (Catell, 1997) were developed to cover the missing object oriented elements. In the case of ODL, a class description can contain the following elements: attributes, methods, inheritance parameters, visibility, relationships and integrity rules. These models provide a powerful complexity for software engineering but they are not very flexible to describe data models of higher abstraction.

Global investigations were focused on the SDM with simpler and more universal elements. The most widely known high level semantic models are semantic networks and ontology models. A semantic network is represented with a directed graph where the vertices are the concepts and the edges are the relationships. The main differences between ontology models and the traditional SDM are in the followings: there is no fixed structural hierarchy among the concepts, flexible relationships, independence from application domain, structure is mapped into a logical formula, it can be related to an inference engine. It is widely assumed that anything at a high level of information processing must be based on ontology (Sloman, 2003). Further details can be found on current applications of ontology among others in (Taniar, 2006).

One of the first languages for ontology is RDF (Lassila, 1999). RDF is used to describe the concepts in a neutral, machine-readable format. According to the specification, the basic language elements are resources, literals and statements. There are two types of resources: entity resources and properties. A statement is a triplet (p,s,o), where p is a property, s is a resource and o is either a literal or a resource. In another approach, p is called predicate, s is the subject and o is the object in the statement. As it can be seen, a statement corresponds to an information atom.

A pioneer representative of the next generation of languages is OWL (Bechhofer, 2004) which can be considered as an extension of RDF, that contains extra elements to describe among others typing, property characteristics, cardinality and behavioral properties. The OWL-DL language is based on Description Logic that describes the structural relationships of the domain in a logic language, which enables automatic reasoning and constraint checking in the system. The applied logic language is based on first-order predicate logic. The most widely used products related to OWL are Protégé, Pellet and KAON2.

MULTI-LAYERED SEMANTIC MODELS

Multi-Layered Schemas

In the case of systems with complex functionality, one way to reduce complexity is to build up a modular system. Modularization is a successful concept in all engineering areas. Modularization can be vertical or horizontal. Vertical modularization is called layering. The basic properties of a layered system are the followings:

- the elements are assigned to clusters (called layers);
- there exists a hierarchical relationship between the clusters;
- the relationships within the clusters differ from the relationships between the clusters;
- the clusters cooperate with each other in the role of a client or of a server.

Every layer offers a set of functionality where the functions are built upon the services of the underlying layers. In the case of a multi-layered system, the implementation can gain in cost reduction compared with a single-layer structure. Layering means modularization from the viewpoint of implementation and it has the following qualitative and quantitative benefits (Knoerschild, 2003):

- encapsulation (the layers are in great part self-contained, consistency) ,
- independence,
- flexibility (the layers can be replaced without affecting the other layers),
- cost reduction (simplicity in testing and in design, reusability).

The layered structure is a common technology nowadays among others in networking (Hnatyshin, 2007), image processing (Sunitha, 2007), process control (Zender, 2007) and software development (Kreku, 2006).

Multi-Layered Nature of Human Recognition

It was realized very early that human spatial cognition is based on a partially hierarchical conceptual view of space (McNamara, 1986). It is usual to perform the mapping of spatial environment with a semantic hierarchy (Kuipers, 2000). In the proposal of Sloman (2003), the internal representation of the spatial environment is implemented with a three-layered model. The lowest layer is called metric layer. It establishes an absolute frame of reference. It consists of a navigation graph that describes the important positions of the environment. In the next, topological layer, the navigational nodes are mapped into areas, where an area corresponds to a set of connected nodes. An area denotes a compound spatial concept. The highest level belongs to the conceptual layer. In this layer the areas are mapped to general abstract concepts. This level corresponds to the ontology layer that provides different relationships and a reasoning engine.

According to the current H–Cogaff view of human information processing architecture (Sloman, 2003) the cognition system consists of several regions performing concurrent activities. The regions are structured into hierarchies. The perception hierarchy can activate for example different concepts at the same time for a single sensor input image. Visual perception can detect different levels of structure and different levels of concepts. The developed multi-layer ontology model consists of three layers: the reactive, the deliberate and the meta-management layer.

Also in artificial intelligence, the application of multi-layered structures has gained a larger popularity. In (Kamimura, 2003), the information theoretical competitive learning method was implemented with multi-layered networks to solve complex problems. Networks are composed of several competitive layers. In each competitive layer, information is maximized. This successive information maximization enables networks to extract features gradually. Experimental results confirmed that information can be maximized in multi-layered networks, and the networks can extract features that cannot be detected by single-layered networks

Multi-Layered Concept Models

Traditional SDM models were intended to manage only single-layer structure. Neither of the original versions of ER, RDF or OWL uses layers in the model. On the other hand, it can be seen, that the layered structure has a lot of benefits:

- increase in simplicity of management,
- decrease in complexity,
- increase in flexibility, and
- increase in reusability.

The first classic layered models for semantic networks were developed in the 80's. The strong influence of psychological theories in human cognition can be easily observed in the proposed models. The layered model of Thompson (1990) consists of five layers similar in Greenwald's (1988) model. The base layer represents sensor data (images, sounds, signs) and the temporal relationships between these data items. The next layer is devoted to basic concepts. The connection of a concept and of its sensory appearances may vary and may be very complex (transformations). This connection should perform more complex activity than just simple association. The next level is called the level of events. In this layer, the simple object instances are bound to series and sentences. This layer should contain the logic supporting ideas of time and causality. The next level generates abstract objects that cluster object instances together. The next model level describes the activities on the abstract concepts like planning and modeling. The highest level is related to the abstract concepts and abstract activities like reasoning and metadata management.

In the proposal of Khosla (2004), the layering of ontology is strongly correlated with the functional layering of the system. The paper describes the structure of a general soft-computing module. The most internal layer is the object layer to describe the data schema. On the top of the data layer are the distributed agent layer, the tool agent layer and the optimization agent layer. These layers perform among others preprocessing, transformation and decision making. The concept of multi-layering can be applied to the traditional data models, too. A layered UML model is represented

M

among others in (Kreku, 2006). The layers here also correspond to the different functional areas within the application. The three proposed layers are the component layer, the HW architecture layer and the platform architecture layer.

In (Sunitha, 2007), the investigation is focused on the SDM part only. The semantic data model is divided into three layers. The bottom part is the concept measure layer that contains the descriptions of the concepts themselves. The middle layer is used to store the relationships (like specialization, classification) among the concepts. The top layer is for the context-related knowledge elements describing the environment of the application field.

In some other proposals, layering refers not to the functional structure but to the abstraction levels. In the classical UML (Terrasse, 2001), a four-layered metamodel architecture is used. The bottom layer is the object layer and the next layer is the model layer. On the top of the model layer is the metamodel layer. At the top, the meta-metamodel layer can be found. The metaobject facility (MOF) model is based on a layered, conceptual metamodel structure. The content of a conceptual layer describes the elements in the next layer down. Both UML and MOF are based on class oriented representations. In (Melnik, 2000), a three-layer abstraction model is defined for the semantic web. These layers are the syntax layer, the object layer and the semantic layer.

The semantic data model presented in (Sieber, 2006) was invented as a model for bridging the gap between data and semiotics in terms of Peirce with a special focus on technical documentation processes. The concerned knowledge can be shared between many people in different departments, different production locations (including different countries) and in different applications. As a consequence of this, in such a process nearly everyone has two roles: one of owning and dating knowledge; and one of searching and needing dated knowledge. For mathematical purposes this model was extended (Sieber, 2007) using a semantic network based upon a lattice of concepts. The result is a multi-layer semantic data model that can be used to visualize more general decoding and encoding processing between signals and (semantic) concepts.

FUTURE TRENDS

The term of layered ontology occurs very rarely in the literature. The main reason for this is that the basic ontology can describe any levels of abstractions. Thus a single layer can cover any levels of concepts. This monolithic structure will cause difficulties in integrating existing ontologies, as the overlap of concepts is harder to detect. The current research projects on ontology are usually aimed at the development of accurate ontologies for the different application domains. The main current areas are: medical systems, geographical systems, linguistics, social studies, enterprise information systems, logic, knowledge representation and automatic reasoning. Very few proposals deal with the application of modular, multi-layered ontologies. In (Purao, 2005), for example a database design specific domain is analyzed, where three layers are defined: the core (local) level, the neighborhood level and the global domain level.

Although the importance of a domain independent ontology is visible and clear for everybody, the current works seem to neglect this requirement. According to (Mikroyannidis, 2006), ontology management in most information systems is based on simplicity, ontology layering is rarely used, and the requirements for ontology evolution and integration are usually neglected. We can predict that the modularized, layered ontology models will get more attention in the near future.

CONCLUSION

Traditional semantic models are based on a single-layer structure. This approach is a drawback in the development of complex systems. As the model of human cognition is built on a multi-layer approach, and the goal of semantic models is to describe concepts of our world, the multi-layer models seem to be more accurate to create a global semantic model. The current semantic models, like UML or ontology provide some layering possibilities, but the detailed analysis of multi-layered semantic models is a task of the future.

REFERENCES

Bechhofer,. S. et al (2004). OWL Web Ontology Language Reference, homepage:http://www.w3.org/TR/2004/rec-owL-REF-20040210

Catell, R. et all. (1997). The Object Database Standard: ODMG 2.0, The Morgan Kaufmann Publisher, ISBN 15589604634

Chen, P. (1976). The Entity-Relationship Model: Toward a unified view of data, ACM on Database Systems, pp. 9-36

Greenwald, A. (1988). Levels of representation, Technical paper, homepage: http://faculty.washington.edu/agg/unpublished.htm

Gustavsson L., Torgersson O.(2005): Benefits of Multi-Layer Design in Software with Multi-User Interfaces- A Three Step Study, Proceedings of Software Engineering

Hnatishin, V. & Gramatges, G. & Stiefel, M. (2007): Practical Considerations for Extending Network Layer Models with OPNET Modeler, Proceedings of Modeling and Simulations

Kamimura, R. (2003). **Information theoretic competitive learning in self-adaptive multi-layered networks**, Connection Science, Vol. 15., pp. 3-26.

Khosla, R. (2004). Multi-layered Distributed Agent Technology for Soft computing systems, International Journal Knowledge-based Intelligent Information and Engineering Systems , Vol 8. , pp. 117-128.

Knoerschild, K. (2003). A Layering challenge, Technical paper, homepage: http://www.artima.com/weblogs/viewpost.jsp/thread=88316

Kreku, J. et al. (2006). Layered UML Workload and SystemC Platform Models for Performance Simulation, Proceedings of FDL2006

Kuipers, B. (2000), The Spatial Semantic Hierarchy, Artificial Intelligence Vol. 119, pp. 191-233

Lassila, O. & Swick, R. (1999). Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, homepage: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

McNamara, T. (1986). Mental representations of spatial relations, Cognitive Psychology, Vol 18, pp. 87-121.

Mikroyannidis, A. & Theodouldis, B.(2006): Heraclitus II: A Framework for Ontology Management and Evolution, Proceedings of IEEE/WIC/ACM Conference on Web Intelligence

Melnik, S. & Decker, S. (2000). A Layered Approach to Information Modeling and Interoperability on the WEB, Proceedings. of ECDL Workshop on the Semantic Web, pp. 10-23.

Purao, S. & Storey, V. (2005). A multi-layered ontology for comparing relationship semantics, Applied Ontology, I, pp. 117-139

Sieber, T. & Kammerer, M.. (2006). Daten, Wissen und Information. Eine Grundlagenanalyse unter besonderer Berücksichtigung der technischen Dokumentation, source: www.iit.uni-miskolc.hu/~kovacs

Sieber, T. & Kovacs, L.. (2007). Development of a multi-layer concept network, Technical paper, source: www.iit.uni-miskolc.hu/~kovacs

Sloman, A. (2003). What kind of virtual machine is capable of human consciousness, Proceedings of ECAP2003

Sloman, A.(2003). Human vision – a multi-layered multi-functional system, Proceedings of BMVA2003

Sunitha, A.& Govindarajului, P. (2007). Multilayer Semantic data Model for Sports Video, IJCSN Journal, Vol 7, pp. 330-341

Taniar, D. (2006). WEB Semantics and Ontology, Igi Global Publisher, ISBN 9781591409069

Terrasse, M., Savonnet, M. & Becker, G. (2001). An UML-metamodeling Architecture for Interoperability of Information Systems, Proceedings of IDEAS01

Thalheim, B. (2000). Entity-Relationship Modeling (EER), Springer Verlag

Thompson, I. (1990). Layered Cognitive Networks, Generative Science, homepage: www.generativescience.org/ps-papers/layer7.html

WO (2007). Webopedia database, homepage: http://www.webopedia.com/TERM/d/data.html

Zender, H. & Kruijff, G. (2007). Multi-layered conceptual spatial mapping for autonomous mobile robots, Proc. of Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems, pp. 62-66.

KEY TERMS

Data Model: A formal description language to describe and to manipulate the investigated data instances. It contains three components: a static structural part, an integrity part and a manipulation part.

Multi-Layered Data Model: A data model where the model elements are assigned to levels. In the model, a hierarchy is defined between the levels. Regarding the element-level relationships, the intra-level relationships differ from the inter-level relationships.

Ontology: A semantic data model that describes the concepts and their relationships. It contains a controlled vocabulary and a grammar for using the vocabulary terms. The ontology enables to make queries and assertions and reasoning. The most popular form to describe ontology is RDF and OWL.

OWL: A language to describe Web ontologies. It uses an XML format and it contains a formal description logic component, too. It provides the following extra functionality: classification, type and cardinality constraints, thesauri, decidability.

RDF: A semantic data model that describes the world with statements. A statement is a triplet having the following form: subject-predicate-object.

Semantic Data Model: A high level data model. It is usually based on concepts and it uses a graphical formalism. It contains only the key, the semantic properties of the data structure. It does not cover the details of the implementation.

UML: A standardized general-purpose modeling language for object oriented software systems. It has a graphical notation and contains several diagrams: structure diagrams (class, object, component, package) and behavioral diagrams (activity, use-case, state machine, interaction).

M

Multilogistic Regression by Product Units

P. A. Gutiérrez
University of Córdoba, Spain

C. Hervás
University of Córdoba, Spain

F. J. Martínez-Estudillo
INSA – ETEA, Spain

M. Carbonero
INSA – ETEA, Spain

INTRODUCTION

Multi-class pattern recognition has a wide range of applications including handwritten digit recognition (Chiang, 1998), speech tagging and recognition (Athanaselis, Bakamidis, Dologlou, Cowie, Douglas-Cowie & Cox, 2005), bioinformatics (Mahony, Benos, Smith & Golden, 2006) and text categorization (Massey, 2003). This chapter presents a comprehensive and competitive study in multi-class neural learning which combines different elements, such as multilogistic regression, neural networks and evolutionary algorithms.

The Logistic Regression model (LR) has been widely used in statistics for many years and has recently been the object of extensive study in the machine learning community. Although logistic regression is a simple and useful procedure, it poses problems when is applied to a real-problem of classification, where frequently we cannot make the stringent assumption of additive and purely linear effects of the covariates. A technique to overcome these difficulties is to augment/replace the input vector with new variables, basis functions, which are transformations of the input variables, and then to use linear models in this new space of derived input features. Methods like sigmoidal feed-forward neural networks (Bishop, 1995), generalized additive models (Hastie & Tibshirani, 1990), and PolyMARS (Kooperberg, Bose & Stone, 1997), which is a hybrid of Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991) specifically designed to handle classification problems, can all be seen as different non-linear basis function models. The major drawback of these approaches is stating the typology and the optimal number of the corresponding basis functions.

Logistic regression models are usually fit by maximum likelihood, where the Newton-Raphson algorithm is the traditional way to estimate the maximum likelihood a-posteriori parameters. Typically, the algorithm converges, since the log-likelihood is concave. It is important to point out that the computation of the Newton-Raphson algorithm becomes prohibitive when the number of variables is large.

Product Unit Neural Networks, PUNN, introduced by Durbin and Rumelhart (Durbin & Rumelhart, 1989), are an alternative to standard sigmoidal neural networks and are based on multiplicative nodes instead of additive ones.

BACKGROUND

In the classification problem, measurements x_i, $i = 1,2,...,k$, are taken on a single individual (or object), and the individuals are to be classified into one of J classes on the basis of these measurements. It is assumed that J is finite, and the measurements x_i are random observations from these classes. A training sample $D = \{(\mathbf{x}_n, \mathbf{y}_n); n = 1, 2,...,N\}$ is available, where $\mathbf{x}_n = (x_{1n},...,x_{kn})$ is the vector of measurements taking values in $\Omega \subset \mathbb{R}^k$, and $\mathbf{y}_n$ is the class level of the nth individual. In this chapter, we will adopt the common technique of representing the class levels using a "1-of-J" encoding vector $\mathbf{y} = (y^{(1)}, y^{(2)},...,y^{(J)})$, such as $y^{(l)} = 1$ if $\mathbf{x}$ corresponds to an example belonging to class l and $y^{(l)} = 0$ otherwise. Based on the training sample, we wish to find a decision function $C : \Omega \to \{1,2,...,J\}$ for classifying the individuals. In other words, C provides a partition, say $D_1, D_2,...,D_J$, of Ω, where D_l corresponds to the lth class, $l = 1,2,...,J$,

and measurements belonging to D_l will be classified as coming from the *l*th class. A misclassification occurs when a decision rule *C* assigns an individual (based on measurements vector) to a class *j* when it is actually coming from a class $l \neq j$.

To evaluate the performance of the classifiers we can define the Correctly Classified Rate by

$$CCR = \frac{1}{N} \sum_{n=1}^{N} I(C(\mathbf{x}_n) = \mathbf{y}_n) ,$$

where $I(.)$ is the zero-one loss function. A good classifier tries to achieve the highest possible *CCR* in a given problem.

Suppose that the conditional probability that $\mathbf{x}$ belongs to class *l* verifies:

$$p\left(y^{(l)} = 1 \middle| \mathbf{x}\right) > 0, \ \ l = 1, 2, ..., J, \ \mathbf{x} \in \Omega ,$$

and set the function:

$$f_l(\mathbf{x}, \boldsymbol{\theta}_l) = \log \frac{p\left(y^{(l)} = 1 \middle| \mathbf{x}\right)}{p\left(y^{(J)} = 1 \middle| \mathbf{x}\right)}, \ \ l = 1, 2, ..., J, \ \mathbf{x} \in \Omega$$

where $\boldsymbol{\theta}_l$ is the weight vector corresponding to class *l* and $f_J(\mathbf{x}, \boldsymbol{\theta}_J) \equiv 0$. Under a multinomial logistic regression, the probability that $\mathbf{x}$ belongs to class *l* is then given by

$$p\left(y^{(l)} = 1 \middle| \mathbf{x}, \boldsymbol{\theta}\right) = \frac{\exp f_l\left(\mathbf{x}, \boldsymbol{\theta}_l\right)}{\sum\limits_{j=1}^{J} \exp f_j\left(\mathbf{x}, \boldsymbol{\theta}_j\right)}, \ l = 1, 2, ..., J$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{J-1})$.

The classification rule coincides with the optimal Bayes' rule. In other words, an individual should be assigned to the class which has the maximum probability, given the vector measurement $\mathbf{x}$:

$$C(\mathbf{x}) = \hat{l} ,$$

where

$$\hat{l} = \arg\max_l f_l(\mathbf{x}, \hat{\boldsymbol{\theta}}_l), \ \text{for } l = 1, ..., J.$$

On the other hand, because of the normalization condition we have:

$$\sum_{l=1}^{J} p\left(y^{(l)} = 1 \middle| \mathbf{x}, \boldsymbol{\theta}\right) = 1 ,$$

and the probability for one of the classes (in the proposed case, the last) need not be estimated (observe that we have considered $f_J(\mathbf{x}, \boldsymbol{\theta}_J) \equiv 0$).

MULTILOGISTIC REGRESSION AND PRODUCT UNIT NEURAL NETWORKS

Multilogistic Regression by using Linear and Product-Unit models (MLRPU) overcomes the nonlinear effects of the covariates by proposing a multilogistic regression model based on the combination of linear and product-unit models, where the nonlinear basis functions of the model are given by the product of the inputs raised to arbitrary powers. These basis functions express the possible strong interactions between the covariates, where the exponents are not fixed and may even take real values. In fitting the proposed model, the non-linearity of the PUNN implies that the corresponding Hessian matrix is generally indefinite and the likelihood has more local maximum. This reason justifies the use of an alternative heuristic procedure to estimate the parameters of the model.

Non-Linear Model Proposed

The general expression of the proposed model is given by:

$$f_l(\mathbf{x}, \boldsymbol{\theta}_l) = \alpha_0^l + \sum_{i=1}^{k} \alpha_i^l x_i + \sum_{j=1}^{m} \beta_j^l \prod_{i=1}^{k} x_i^{w_{ji}}, \ \ l = 1, 2, ..., J-1$$

where

$$\boldsymbol{\theta}_l = (\boldsymbol{\alpha}^l, \boldsymbol{\beta}^l, \mathbf{W}),$$

$$\boldsymbol{\alpha}^l = (\alpha_0^l, \alpha_1^l, ..., \alpha_k^l),$$

$$\boldsymbol{\beta}^l = \left(\beta_1^l, ..., \beta_m^l\right) \text{ and}$$

$$\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m),$$

with $\mathbf{w}_j = (w_{j1}, w_{j2},..., w_{jk})$,

$w_{ji} \in \mathbb{R}$.

As has been stated before, the nonlinear part of $f_l(\mathbf{x},\boldsymbol{\theta}_l)$ corresponds to Product Unit Neural Networks (PUNN), introduced by Durbin and Rumelhart (Durbin & Rumelhart, 1989) and subsequently developed by other authors (Janson & Frenzel, 1993), (Leerink, Giles, Horne & Jabri, 1995), (Ismail & Engelbrecht, 2000), (Martínez-Estudillo, Hervás-Martínez, Martínez-Estudillo & García-Pedrajas, 2006), (Martínez-Estudillo, Martínez-Estudillo, Hervás-Martínez & García-Pedrajas, 2006). Advantages of product-unit based neural networks include increased information capacity and the ability to form higher-order combinations of inputs. They are universal approximators and it is possible to obtain upper bounds of the VC dimension of product unit neural networks that are similar to those obtained for sigmoidal neural networks (Schmitt, 2001). Despite these obvious advantages, product-unit based neural networks have a major drawback. Their training is more difficult than the training of standard sigmoidal based networks (Durbin & Rumelhart, 1989). The main reason for this difficulty is that small changes in the exponents can cause large changes in the total error surface. Hence, networks based on product units have more local minima and a greater probability of becoming trapped in them. It is well-known (Janson & Frenzel, 1993) that back-propagation is not efficient in training product-units. Several efforts have been made to develop learning methods for product units (Leerink, Giles, Horne & Jabri, 1995), (Martínez-Estudillo, Martínez-Estudillo, Hervás-Martínez, & García-Pedrajas, 2006), mainly in a regression context.

Estimation of the Model Coefficients

In the supervised learning context, the components of the weight vectors $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2,..., \boldsymbol{\theta}_{J-1})$ are estimated from the training dataset D. To perform the maximum likelihood (ML) estimation of $\boldsymbol{\theta}$, one can minimize the negative log-likelihood function

$$L(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N}\log p\left(\mathbf{y}_n \big| \mathbf{x}_n, \boldsymbol{\theta}\right) =$$

$$= \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{l=1}^{J}y_n^{(l)}f_l(\mathbf{x}_n,\boldsymbol{\theta}_l) + \log\sum_{l=1}^{J}\exp f_l(\mathbf{x}_n,\boldsymbol{\theta}_l)\right]$$

The error surface associated with the proposed model is very convoluted with numerous local optimums. The non-linearity of the model with respect to the parameters $\boldsymbol{\theta}_l$ and the indefinite character of the associated Hessian matrix do not recommend the use of gradient-based methods to maximize the log-likelihood function. Moreover, the optimal number of basis functions of the model (i.e. the number of hidden nodes in the product-unit neural network) is unknown. Thus, the estimation of the vector parameter $\hat{\boldsymbol{\theta}}$ is carried out by means a hybrid procedure described below.

In this paragraph we make a detailed description of the different aspects of the MLRPU methodology. The process is structured in four steps:

Step 1. We apply an evolutionary neural network algorithm to find the basis functions

$$B(\mathbf{x}, \hat{\mathbf{W}}) = \left\{B_1(\mathbf{x}, \hat{\mathbf{w}}_1), B_2(\mathbf{x}, \hat{\mathbf{w}}_2),..., B_m(\mathbf{x}, \hat{\mathbf{w}}_m)\right\}$$

corresponding to the nonlinear part of $f(\mathbf{x},\boldsymbol{\theta})$. We have to determine the number of basis functions m and the weight vector $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2,..., \mathbf{w}_m)$.

To apply evolutionary neural network techniques, we consider a PUNN with the following structure (Fig. 1): an input layer with a node for every input variable, a hidden layer with several nodes, and an output layer with one node for each category. The activation function of the j-th node in the hidden layer is given by

$$B_j(\mathbf{x},\mathbf{w}_j) = \prod_{i=1}^{k} x_i^{w_{ji}}$$

where w_{ji} is the weight of the connection between input node i and hidden node j and $\mathbf{w}_j = w_{j1},..., w_{jk})$. The activation function of the output node l is given by

$$g_l(\mathbf{x},\boldsymbol{\beta}^l,\Omega) = \beta_0^l + \sum_{j=1}^{m}\beta_j^l B_j(\xi,\omega_\mathbf{j})$$

where β_j^l is the weight of the connection between the hidden node j and the output node l. The transfer function of all output nodes is the identity function.

The weight vector $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2,..., \mathbf{w}_m)$ is estimated by means of an evolutionary programming algorithm detailed in (Hervás-Martínez, Martínez-Estudillo & Gutiérrez, 2006), that optimizes the error function

Figure 1. Model of a product-unit based neural network
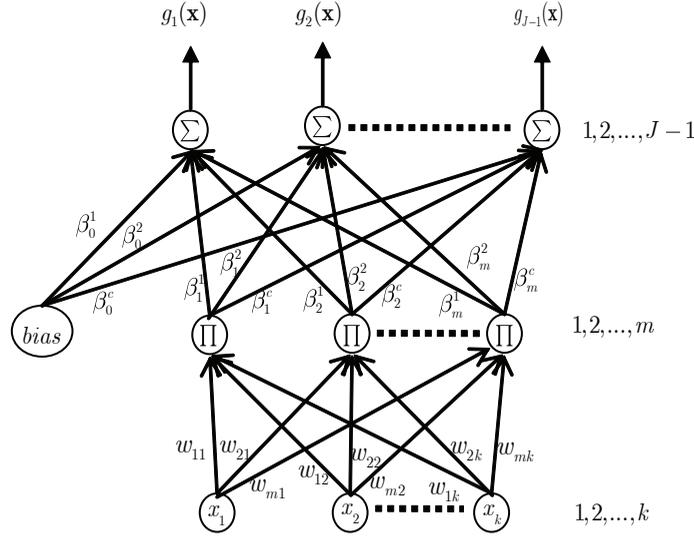
given by the negative log-likelihood for *N* observations associated with the product-unit model:

$$L^*(\boldsymbol{\beta}, \mathbf{W}) = \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{l=1}^{J-1} y_n^{(l)} g_l(\mathbf{x}_n, \boldsymbol{\beta}^l, \mathbf{W}) + \log\sum_{l=1}^{J-1}\exp g_l(\mathbf{x}_n, \boldsymbol{\beta}^l, \mathbf{W}) \right]$$

Although in this step the evolutionary process obtains a concrete value for the $\boldsymbol{\beta}$ vector, we only consider the estimated weight vector $\hat{\mathbf{W}} = (\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, ..., \hat{\mathbf{w}}_m)$ that builds the basis functions. The value for the $\boldsymbol{\beta}$ vector will be determined in Step 3 together with the $\boldsymbol{\alpha}$ coefficient vector.

Step 2. We consider the following transformation of the input space by adding the nonlinear basis functions obtained by the evolutionary algorithm in step 1:

$$H : \mathbb{R}^k \to \mathbb{R}^{k+m}$$
$$(x_1, x_2, ..., x_k) \to (x_1, x_2, ..., x_k, z_1, ..., z_m)$$

where $z_1 = B_1(\mathbf{x}, \hat{\mathbf{w}}_1), ..., z_m = B_m(\mathbf{x}, \hat{\mathbf{w}}_m)$.

Step 3. We minimize the negative log-likelihood function for *N* observations:

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{l=1}^{J} y_n^{(l)}(\boldsymbol{\alpha}^l\mathbf{x}_n + \boldsymbol{\beta}^l\mathbf{z}_n) + \log\sum_{l=1}^{J}\exp(\boldsymbol{\alpha}^l\mathbf{x}_n + \boldsymbol{\beta}^l\mathbf{z}_n) \right]$$

where $\mathbf{x}_n = (1, x_{1n}, ..., x_{kn})$. Now, the Hessian matrix of the negative log-likelihood in the new variables $x_1, x_2, ..., x_k, z_1, ..., z_m$ is semidefinite positive. Then, we could apply Newton's method, also known, in this case, as Iteratively Reweighted Least Squares (IRLS). Although there are other methods for performing this optimization, none clearly outperforms IRLS (Minka, 2003). The estimated vector coefficient $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\mathbf{W}})$ determines the model:

$$f_l(\mathbf{x}, \hat{\boldsymbol{\theta}}) = \hat{\alpha}_0^l + \sum_{i=1}^{k}\hat{\alpha}_i^l x_i + \sum_{j=1}^{m}\hat{\beta}_j^l \prod_{i=1}^{k} x_i^{\hat{w}_{ji}},$$

$$l = 1, 2, ..., J-1$$

Step 4. In order to select the final model, we use a backward stepwise procedure, starting with the full model with all the covariates, initial and PU covariates, and successively prune variables sequentially to the model until further pruning does not improve the fit.

Application to *Remote Sensing*

We have tested the proposed methodology in a real agronomical problem of precision farming, consisting of mapping weed patches in crop fields, through remote sensed data.

Remote sensing systems can provide a large amount of continuous field information at a reasonable cost. Remote sensed imagery shows great potential in modelling different agronomic parameters for its application in precision farming. One aspect of overcoming the possibility of minimizing the impact of agriculture on environmental quality is the development of more efficient approaches for crop production determination and for site-specific weed management. Potential economic and environmental benefits of site-specific herbicide applications include reduced spray volume, herbicide costs and non-target spraying, and increased control of weeds, (Thompson, Stafford & Miller, 1991), (Medlin, Shaw, Gerard, & LaMastus, 2000).

We face a mapping weed patches problem through the analysis of aerial photographs. Images and data sets have been given by the Precision Farming and Remote Sensing Unit of the Department of Crop Protection, Institute of Sustainable Agriculture (CSIC, Spain), whose members reported previous results in predicting *Ridolfia segetum* Moris patches, (Peña-Barragán, López-Granados, Jurado-Expósito & García-Torres, 2007). The data analyzed correspond to a study conducted in 2003 at the 42 ha-farm Matabueyes, naturally infested by *R. segetum*. At a field study, the nature of 2,400 pixels was determined, being them considered as ground-truth pixels: 800 pixels were classified as *R. segetum* and 800 pixels were classified as *R. segetum* free pixels.

Input variables include the digital values of all bands in each available image, that is: Red (R), Green (G) and Blue (B), for June image, and R, G, B and Near Infrared (NIR) for May and July images. The experimental design was conducted using a stratified holdout cross-validation procedure, where the size of training set was approximately $0.7n$ (1,120 pixels) for the training set and $0.3n$ (480 pixels) for the generalization set, n being the size of the full dataset.

In all experiments, the EA has been applied with the same parameters. SPSS 13.0 software (SPSS, 2005) was used for applying IRLS algorithm and in order to select the more significant variables in the final model, through a backward stepwise procedure.

The models compared in the different experiments are the following: firstly, we extract the best PUNN model of the EA (EPUNN); secondly, we obtain standard Logistic Regression model using initial covariables (LR); finally, we apply Logistic Regression only over basic function extracted from EPUNN model (MLRPU) and over the same basic functions together with initial covariables (MLRLPU).

Results

Performance of each model has been evaluated using the Correctly Classified Ratio in the generalization set (CCR_G). In Table 1 we show the matrix results of classification over train and generalization sets for the three classification problems and the four models proposed (one problem at each date, May, June and July, and four models, EPUNN, LR, MLRPU and MLRLPU). Best CCR_G results are obtained with MLRPU and MLRLPU at May and June, although at July EPUNN model yields the best results. At all dates, differences between standard LR and hybrid LR (MLRPU and MLRLPU) are very significant. Table 2 includes models obtained at the date that leads to better classification results, that is, at June.

Using these models we can obtain the probability of *R.segetum* presence at all pixels of the image, including non ground-truth pixels. Figure 1, 2, 3 and 4 represents weed maps obtained using the four proposed models at June, EPUNN, LR, MLRPU and MLRLPU, respectively. Weed presence probability has been represented using a scale between white (minimum probability, nearly 0) and dark green (maximum probability, nearly 1). From these maps, the agronomical expert can decide what threshold probability value consider to apply herbicide. MLRLPU and MLRPU models clearly differentiate better between high weed density zones and weed free zones, so they have a higher interest from the point of view of intelligent site-specific herbicide application.

Table 1. *Classification matrixes (Y=0, R.segetum absence; Y=1, R.segetum presence) at all dates, using best Evolutionary Product Unit Neural Network, EPUNN, Logistic Regression, LR (in italic), Logistic Regression only with Product Units, MLRPU, (in brackets), and Logistic Regression with initial covariables and Product Units, MLRLPU (in squared brackets)*

Phenological Stage (Date)	Ground Truth Response	Training			Generalization		
		Predicted Response			Predicted Response		
		Y=0	Y=1	CCR (%)	Y=0	Y=1	CCR (%)
Vegetative	Y=0	384 *352* (383) [394]	176 *208* (177) [166]	68.5 *62.9* (68.4) [70.4]	164 *148* (164) [168]	76 *92* (76) [72]	68.3 *61.7* (68.3) [70]
(mid-May)	Y=1	133 *171* (136) [141]	427 *389* (424)419	76.2 *69.5* (75.7) [74.8]	65 *69* (67) [69]	175 *171* (173) [171]	72.9 *71.3* (72.1) [71.3]
	CCR (%)			72.4 *66.2* (72.1) [72.6]			**70.6** *66.5* (70.2) **[70.6]**
Flowering	Y=0	547 *529* (547)[552]	13 *31* (13) [8]	97.7 *94.5* (97.7)[98.6]	236 *226* (237)[238]	4 *14* (3)[2]	98.3 *94.2* (98.8)[99.2]
(mid-June)	Y=1	7 *30* (9) [11]	553 *530* (551)[549]	98.8 *94.6* (98.4) [98]	2 *12* (2) [2]	238 228 (238)[238]	99.2 *95* (99.2)[99.2]
	CCR (%)			98.2 *94.6* (98)[98.3]			98.7 *94.6* (99)**[99.2]**
Senescence	Y=0	443 *296* (443) [447]	117 *264* (117) [113]	79.1 *52.9* (79.1) [79.8]	195 *138* (425) [189]	45 *102* (55) [51]	81.2 *57.5* (88.5) [78.8]
(mid-July)	Y=1	105 *131* (111) [117]	455 *429* (449) [443]	81.2 *76.6* (80.2) [79.l]	52 *60* (53) [50]	188 *180* (187) [190]	78.3 *75* (77.9) [79.2]
	CCR (%)			80.1 *64.7* (79.6) [79.5]			**79.8** *66.3* (79.6) [79]

Table 2. *Obtained models at June for the determination of R.segetum presence probability (P) in order to obtain weed patches maps*

Methodology	#coef.	Model
EPUNN	8	$P = 1/(1+\exp(-(-0.424+75.419(V^{4\,633})+0.322(R^{-1.888}) +14.990(A^{3\,496}\,V^{-3.415}))))$
LR	4	$P = 1/(1+\exp(-(-0.694+8.282(A)-63.342(V)-11.402(R))))$
MLRPU	7	$P = 1/(1+\exp(-(-17.227+143.012(V^{4\,633})+0.636(R^{-1.888})+23.021(A^{3\,496}\,V^{-3.415}))))$
MLRLPU	9	$P = 1/(1+\exp(-(18.027+130.674(A)-133.662(V)-29.346(R)+353.147(V^{4\,633})-3.396(B^{3\,496}\,G^{-3.415}))))$

FUTURE TRENDS

Concepts exposed in this chapter offer the possibility of developing new models of multi-class generalized linear regression, by means of considering different types of basis functions (Sigmoidal Units, Radial Basis Functions and Product Units) for the non-linear part of the proposed model. Moreover, future research could include ordinal logistic regression models with different basis functions or probit models with different basis functions.

Figure 1. EPUNN R.segetum presence probability map

Figure 2. LR R.segetum presence probability map
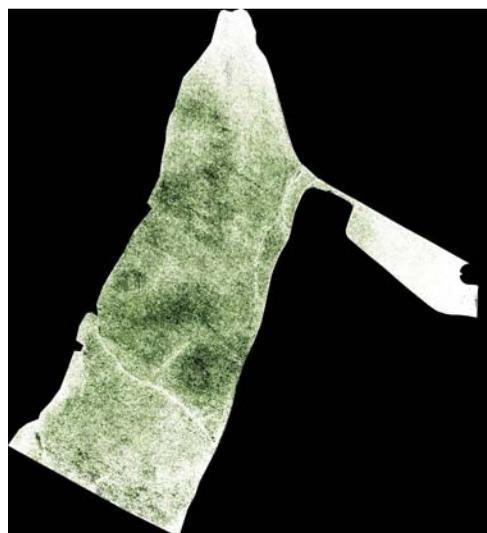
Figure 3. MLRPU R.segetum presence probability map

Figure 4. MLRLPU R.segetum presence probability map

M
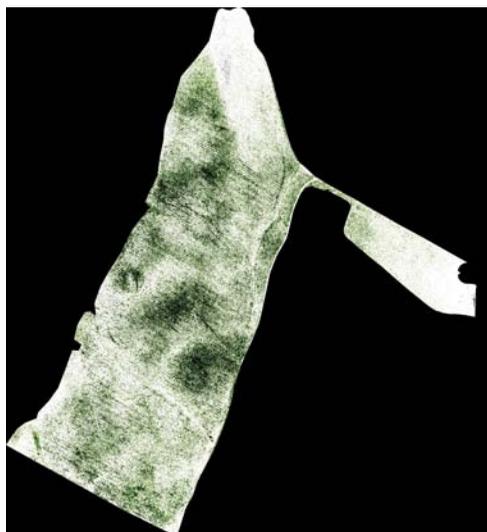
CONCLUSION

To the best of our knowledge, the approach presented in this paper is a study in multi-class neural learning which combines three tools used in machine learning research: the logistic regression, the product-unit neural network model and the evolutionary neural network paradigm. Logistic regression is a well-tested statistical approach that performs well in two-class classification and can naturally be generalized to the multi-class case. On the other hand, product-unit neural network models are an alternative to standard sigmoidal neural networks with the ability to capture non-linear interaction between the input variables. Finally, evolutionary artificial neural networks present an interesting platform for optimizing both network performance and architecture simultaneously. The adequate combination of these three elements carried out in several steps in our proposal, provides a competitive methodology to solve classification problems.

REFERENCES

Athanaselis, T., Bakamidis, S., Dologlou, I., Cowie, R., Douglas-Cowie, E., & Cox, C. (2005). ASR for emotional speech: Clarifying the issues and enhancing performance. *Neural Networks*, 18(4), 437-444.

Bishop, M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press.

Chiang, J.H., (1998). A Hybrid Neural Model in Handwritten Word Recognition. *Neural Networks*, 11(2), 337-346.

Durbin, R., & Rumelhart, D. (1989). Products Units: A computationally powerful and biologically plausible extension to back propagation networks. *Neural Computation*, 1, 133-142.

Friedman, J. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1), 1-141.

Hastie, T., Tibshirani, R.J., & Friedman, J. (2001). *The Elements of Statistical Learning. Data mining, Inference and Prediction*, Springer-Verlag.

Hervás-Martínez, C., Martínez-Estudillo, F.J., Gutiérrez, P.A. (2006). Classification by means of evolutionary product-unit neural networks. *IEEE International Joint Conference on Neural Networks (IJCNN06, Vancouver)*, 2834-2841.

Ismail, A., & Engelbrecht, A.P. (2000). Global Optimization Algorithms for Training Product Unit Neural Networks. *International Joint Conference on Neural Networks*, 1, 132-137.

Janson, D.J., & Frenzel, J.F. (1993). Training product unit neural networks with genetic algorithms. *IEEE Expert: Intelligent Systems and Their Applications*, 8(5), 26-33.

Kooperberg, C., Bose, S., & Stone, C.J. (2006). Polychotomous regression. *Journal of the American Statistical Association*, 92, 117-127.

Leerink, L., Giles, C., Horne, B., & Jabri, M. (1995). Learning with product units. *Advances in Neural Information Processing Systems*, 7, 537-544.

Mahony, S., Benos, P.V., Smith, T.J., & Golden, A. (2006). Self-organizing neural networks to support the discovery of DNA-binding motifs. *Neural Networks*, 19(6-7), 950-962.

Martínez-Estudillo, A.C., Martínez-Estudillo, F.J., Hervás-Martínez, C., & García-Pedrajas, N. (2006). Evolutionary Product Unit based Neural Networks for Regression. *Neural Networks*, 19(1), 477-486.

Martínez-Estudillo, A.C., Hervás-Martínez, C., Martínez-Estudillo, F.J., & García-Pedrajas, N. (2006). Hybridation of evolutionary algorithms and local search by means of a clustering method. *IEEE Transaction on Systems, Man and Cybernetics, Part. B: Cybernetics*, 36(3), 534-546.

Medlin, C.R., Shaw, D.R., Gerard, P.D., & LaMastus, F.E. (2000). Using remote sensing to detect weed infestations in Glycine max. *Weed Science*, 48(3), 393-398.

Minka, T.P., (2003). A comparison of numerical optimizers for logistic regression. http://research.microsoft.com/~minka/papers/logreg/

Peña-Barragán, J.M., López-Granados, F., Jurado-Expósito, M., & García-Torres L. (2007). Mapping Ridolfia segetum patches in sunflower crop using remote sensing. Weed Research, 47(2), 164-172.

Schmitt, M. (2001). On the Complexity of Computing and Learning with Multiplicative Neural Networks. *Neural Computation*, 14(24), 241-301.

SPSS (2005). Advanced Models. Copyright 13.0 SPSS Inc. Chicago, IL.

Thompson, J.F., Stafford, J.V. & Miller P.C.H. (1991). Potential for automatic weed detection and selective herbicide application. *Crop Protection*, 10, 254-259.

KEY TERMS

Artificial Neural Networks: A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Evolutionary Computation: Computation based on iterative progress, such as growth or development in a population. This population is selected in a guided random search using parallel processing to achieve the desired solution. Such processes are often inspired by biological mechanisms of evolution.

Evolutionary Programming: One of the four major evolutionary algorithm paradigms, with no fixed structure or representation, in contrast with some of the other evolutionary paradigm. Its main variation operator is the mutation.

Iteratively Reweighted Least Squares (IRLS): Numerical algorithm for minimizing any specified objective function using a standard weighted least squares method such as Gaussian elimination. It is widely applied in Logistic Regression.

Logistic Regression: Statistical regression model for Bernoulli-distributed dependent variables. It is a generalized linear model that uses the logit as its link function. Logistic regression applies maximum likelihood estimation after transforming the dependent into a logit variable (the natural log of the odds of the dependent occurring or not).

Precision Farming: Use of new technologies, such as global positioning (GPS), sensors, satellites or aerial images, and information management tools (GIS) to assess and understand in-field variability in agriculture. Collected information may be used to more precisely evaluate optimum sowing density, estimate fertilizers and other inputs needs, and to more accurately predict crop yields.

Product Unit Neural Networks: Alternative to standard sigmoidal neural networks, based on multiplicative nodes instead of additive ones. Concretely, the output of each hidden node is the product of all its inputs raised to a real exponent.

Remote Sensing: Short or large-scale acquisition of information of an object or phenomenon, by the use of either recording or real-time sensing devices that is not in physical or intimate contact with the object (such as by way of aircraft, spacecraft, satellite, or ship).

Multi–Objective Evolutionary Algorithms

Sanjoy Das
Kansas State University, USA

Bijaya K. Panigrahi
Indian Institute of Technology, India

INTRODUCTION

Real world optimization problems are often too complex to be solved through analytical means. Evolutionary algorithms, a class of algorithms that borrow paradigms from nature, are particularly well suited to address such problems. These algorithms are stochastic methods of optimization that have become immensely popular recently, because they are derivative-free methods, are not as prone to getting trapped in local minima (as they are population based), and are shown to work well for many complex optimization problems.

Although evolutionary algorithms have conventionally focussed on optimizing single objective functions, most practical problems in engineering are inherently multi-objective in nature. Multi-objective evolutionary optimization is a relatively new, and rapidly expanding area of research in evolutionary computation that looks at ways to address these problems.

In this chapter, we provide an overview of some of the most significant issues in multi-objective optimization (Deb, 2001).

BACKGROUND

Arguably, Genetic Algorithms (GAs) are one of the most common evolutionary optimization approaches. These algorithms maintain a population of candidate solutions in each generation, called chromosomes. Each chromosome corresponds to a point in the algorithm's search space. GAs use three Darwinian operators – selection, mutation, and crossover to perform their search (Mitchell, 1998). Each generation is improved by systematically removing the poorer solutions while retaining the better ones, based on a fitness measure. This process is called selection. Binary tournament selection and roulette wheel selection are two popular methods of selection. In binary tournament selection,

two solutions, called parents, are picked randomly from the population, with replacement, and their fitness compared, while in roulette wheel selection, the probability of a solution to be picked, is made to be directly proportional to its fitness.

Following selection, the crossover operator is applied. Usually, two parent solutions from the current generation are picked randomly for producing offspring to populate the next generation of solutions. The offspring are created from the parent solutions in such a manner that they bear characteristics from both. The offspring chromosomes are probabilistically subject to another operator called mutation, which is the addition of small random perturbations. Only a few solutions undergo mutation. Evolutionary Strategies (ES) forms another class of evolutionary algorithms that is closely related to GAs and uses similar operators as well.

Particle Swarm Optimization (PSO) is a more recent approach (Clerc, 2005). It is modeled after the social behavior of organisms such as a flock of birds or a school of fish, and thus only loosely classified as an evolutionary approach. Each solution within the population in PSO, called a particle, has a unique position in the search space. In each generation, the position of each particle is updated by the addition of the particle's own velocity to it. The velocity of a particle, a vector, is then incremented towards best location encountered in the particle's own history (called the individual best), as well as the best position in the current iteration (called the global best).

EVOLUTIONARY ALGORITHMS FOR MULTI-OBJECTIVE OPTIMIZATION

Multi-Objective Optimization

When dealing with optimization problems with multiple objectives, the conventional theories of optimality can-

Figure 1. An illustration of convergence and diversity concepts in multi-objective optimization algorithms. The objective functions f_1 and f_2 are to be minimized.
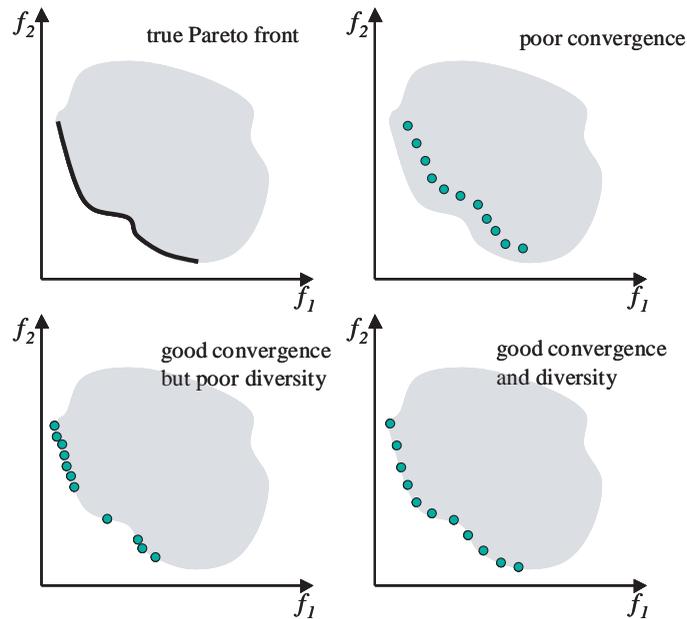
not be applied. Instead, the concepts of dominance and Pareto-optimality are used. Without a loss of generality, we will assume that the optimization problem involves the simultaneous minimization of several objectives only. If these objective functions are $f_i(.)$, $i = 1,...,M$, a solution x is said to *dominate* another solution y if and only if for all i, $f_i(x) \leq f_i(y)$ with at least one of the inequalities being strict. In other words, x dominates y if and only if x is as good as y for all objectives and better in at least one. This relationship is written $x \succ y$. In the set of all feasible solutions, that subset whose members are not dominated by any other in the set, is called the *Pareto set*. In other words, if $\mathbf{S}$ is the search space, the Pareto set $\mathbf{P}$ is given by, $\mathbf{P} = \{x \in \mathbf{S} \mid \forall y \in \mathbf{S}, \ y \succ x \text{ is false}\}$. The image of the Pareto set $\mathbf{P}$ in the M dimensional objective function space is called the *Pareto front*, $\mathbf{F}$. Thus, $\mathbf{F} = \{(f_1(x), f_2(x),... f_M(x)) \mid x \in \mathbf{P}\}$.

The goal of a multi-objective optimization algorithm is twofold. Firstly, its output, the set of non-dominated solutions in the population, must be as close to the true Pareto front as possible. This feature is called *convergence*. Secondly, in addition to good convergence, the multi-objective evolutionary algorithm should also yield solutions that sample the front at approximately regularly spaced intervals, a feature that is usually referred to as *diversity*. Outputs, where the solutions are clustered in a few regions of the front while other regions are either omitted or poorly sampled, are not desirable. Figure 1 illustrates the concepts of good convergence and diversity.

In order to handle multi-objective optimization tasks, an evolutionary algorithm must be equipped to discriminate between solutions using either convergence or diversity as the criterion for comparison. When using convergence, the majority of current evolutionary algorithms make use of one of two basic ranking schemes that were originally put forth by Goldberg (Goldberg, 1989). The first is a method that shall be referred to here as *domination counting*. Within a population of solutions, the rank of any solution is the number of other solutions in the population that dominate it. Clearly, the non-dominated solutions in the population are assigned counts of zero. The second approach will be called *non-dominated sorting*. Here, ranks are assigned to each solution in a population, in such a manner that solutions that have the same rank do not dominate one another, each solution is assigned a lower rank than another that it dominates, and, in turn, is ranked higher

than ones dominating it. As before, non-dominated solutions in the population are assigned a rank of zero. Both concepts are illustrated in figure 2. The numbers corresponding to each solution in figure 2 (left) are the domination counts. The figure shows specifically the solution with a count of 3 being dominated by three others (with ranks 0, 0 and 2). In figure 2 (right), the solutions with equal rank (the ranks being 0, 1 or 2) are grouped together. Solutions with a rank of 0 are non-dominated ones. They dominate those with ranks 1 or 2. When they are removed, those with ranks of 1 are no longer dominated. Removing rank 1 solutions makes the rank 2 solution non-dominated.

Multi-objective evolutionary algorithms must also be equipped with the ability to discern between solutions that are in sparser regions of the M dimensional space of objective functions, from those in denser ones. The three main approaches used to do so are illustrated in figure 3. The first of these methods is to consider a *bounding hypercube* around each solution in the objective function space that does not enclose any other solution (Deb, Pratap, Agarwal & Meyarivan, 2002). Neighboring solutions will be located at some of the corners of this hypercube. This is shown in figure 3 (left), where two solutions, *a* and *b*, have been enclosed by hypercubes. The perimeters of the hypercubes are considered as measures of diversity. Solutions whose bounding hypercubes have a larger perimeter are considered to be located in sparser regions than those with smaller ones. The second approach (Knowles & Corne, 2000) is to superimpose an M-dimensional *hypergrid* in the objective function space, and consider the number of solutions that stay in each of the hypergrid's cells as a measure of how dense the region around the cell is. In figure 3 (middle), since *b* occupies the same cell as another solution, whereas *a* does not, the latter is

regarded as being placed in a sparser region. The last approach computes the kth *nearest neighbor* of each solution (Zitzler, Laumanns & Thiele, 2001). This situation is depicted in figure 3 (right), where the solutions *a* and *b* have been connected to their nearest neighbors ($k=1$). Solutions, which lie at greater distances from their neighbors, are considered to be in sparser regions of the objective function space.

Since elitism, the guaranteed survival of the fittest solutions per generation, shows faster convergence in single objective evolutionary algorithms, this feature has also been incorporated into most current multi-objective evolutionary algorithms. Elitism is ensured by means of an archive that stores the best solutions in each generation. Quite often, the archived solutions are reinserted back into the main population. Archiving is implemented via schemes that we will refer to, collectively, as *elite preservation*.

A Few Recent Evolutionary Algorithms

(*i*) **NSGA-II** (Deb, Pratap, Agarwal & Meyarivan, 2002): NSGA-II (Non-dominated Sorting Genetic Algorithm) maintains a population and a separate archive, each of size N. In each generation, the population is merged with the archive. This merged set is then subject to elite preservation. The new archive is filled by taking the N best-ranked solutions obtained from elite preservation. The same N individuals are also subject to tournament selection, crossover, and mutation to form the population for the next generation.

Elite preservation is implemented in NSGA-II by using non-dominated sorting for convergence, and the hypercube method for diversity. It proposes an $O(MN^2)$ subroutine for fast non-dominated sorting to assign ranks to the solutions in the merged population.

Figure 2. Discriminating between solutions based on convergence
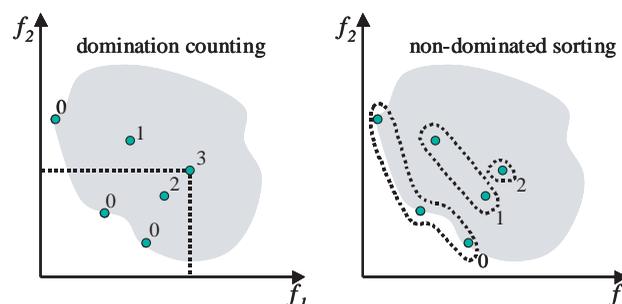
Figure 3. Discriminating between solutions based on diversity
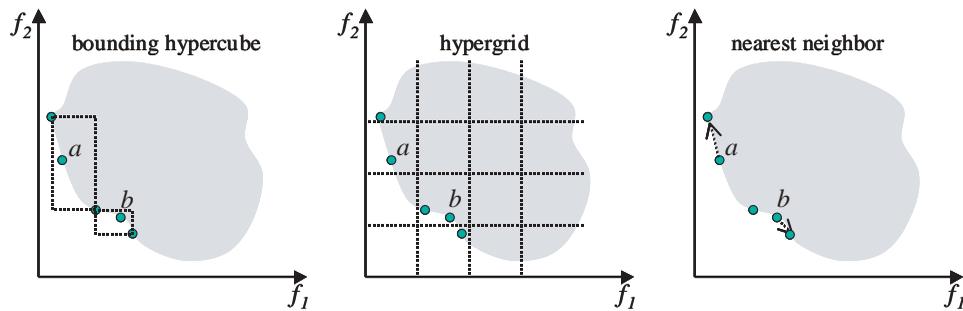
Starting with the lowest ranked solutions, the archive is filled until it reaches its full capacity N.

Diversity is invoked to further discriminate between mutually non-dominating solutions when not all solutions of an identical rank can be inserted into the archive. NSGA-II incorporates an algorithm of order $O(MN \log(N))$ to do so. Because non-dominated sorting is the computationally rate limiting portion in NSGA-II, the overall complexity is $O(MN^2)$ per generation.

(*ii*) **SPEA-2** (Zitzler, Laumanns & Thiele, 2001): The SPEA-2 (Strength Pareto Evolutionary Approach) method is quite similar to NSGA-II. It also maintains a population and an archive of size N, merging both in the beginning of every generation and making use of elite preservation to identify the best to undergo crossover and mutation for the next generation. In SPEA-2, individual fitness are computed in a two-step manner, that is an improved version of the basic domination counting approach discussed earlier. First to be computed is the strength of each solution in the merged population, *i.e.*, the number of solutions it dominates. Then, the raw fitness of each individual is computed as the sum of the strengths of all the solutions that dominate it.

In (Zitzler, Laumanns & Thiele, 2001) it is argued that this method of computing fitness imparts SPEA-2 with some capability to preserve diversity. However, in itself, raw fitness is inadequate to do so, and so a separate term is added to it, that explicitly takes diversity into account. This second term added to each solution's fitness, is inversely related to the distance from the solution to its kth nearest neighbor in objective function space. The overall algorithm complexity of SPEA-2 is $O(MN^3)$ per generation.

(*iii*) **MOPSO** (Coello Coello, 2004): MOPSO (Multi-objective Particle Swarm Optimization) is an approach for fast multi-objective optimization that is based on PSO. It imposes population diversity by means of the M-dimensional hypergrid described earlier, and counting the number of solutions present in each of the hypergrid's cells. Solutions occupying cells with lower counts are preferred to those with higher counts. An even distribution of solutions along the Pareto front is achieved by biasing particles to update their velocities towards global best particles that are located in sparser cells of the hypergrid, *i.e.* with lower counts. This is done using a roulette wheel selection algorithm that picks a cell probabilistically using cell counts, such that the higher the cell count, the lower the probability of selection becomes. MOPSO also implements a mutation operator.

(*iv*) **ParEGO** (Knowles, 2006): ParEGO (Parallel Efficient Global Optimization) has been explicitly designed for problems where evaluating the objective function is highly expensive in terms of computer time. Therefore, ParEGO converges in as few function evaluations as possible. The algorithm uses a Gaussian process model to approximate the fitness landscape that is learned adaptively using supervised learning. For further details one is referred to (Knowles, 2006).

(*v*) **FSGA** (Koduru, Das, Welch & Roe, 2004): FSGA (Fuzzy Simplex Genetic Algorithm) has a complexity of $O(MN^2)$ per generation similar to NSGA-II. It differs from NSGA-II and SPEA-2 in the method used for elite preservation. A measure called fuzzy dominance is used for the purpose. A solution that is not dominated by any other is assigned a fuzzy dominance of zero. The poorer a solution is, the higher the fuzzy dominance value it is assigned. FSGA's fuzzy dominance is a numerical method that not only uses Pareto-optimality, but also considers the degree to which one solution dominates

M

another, making effective use of differences between their values of the objective functions. It has been designed specifically so that FSGA can be hybridized readily with a local search algorithm.

PAES (The Pareto Archive Evolutionary Strategy) and PESA (Pareto Envelope based Selection Algorithm) are other successful multi-objective evolutionary algorithms that make use of the hypergrid to measure diversity (Knowles & Corne, 2000, Corne, Jerram, Knowles & Oates, 2001). Another algorithm, RDGA (Rank Density based Genetic Algorithm) uses this method along with a ranking scheme wherein a non-dominated individual has unit rank and others are assigned one plus the sum of the ranks of all solutions that dominate them (Lu & Yen, 2003). Very recently, the use of fuzzy dominance has been successfully applied to another multi-objective PSO algorithm (Koduru, Das & Welch, 2007).

FUTURE TRENDS

Multi-objective evolutionary optimization is a rapidly expanding, new field of research. Although several interesting approaches have been proposed in the recent literature, further investigation is necessary before multi-objective algorithms can truly address the needs of the application domains.

One current research focus is in devising numerical metrics to compare solutions. This is particularly useful when the problem contains a large number of objectives. In higher dimensional objective function space, it is less likely to find a solution that dominates another, i.e. is better than or equal to another in all objectives. Under these circumstances, comparing solutions that are already within the Pareto front is essential. One such method has been suggested recently (Farina & Amato, 2004). This method counts the number of objectives along which one solution is better than and worse than another, and proposed fuzzy metrics based on the counts. However, such ideas have yet to be incorporated within evolutionary algorithms. A related direction of research is in devising schemes to compare solutions in the presence of uncertainty in objective functions. This research has obvious practical implications in engineering and other applications where measuring objectives such as cost, efficiency or expected lifetime are difficult tasks (Fieldsend, Everson & Singh, 2005).

Another direction of certain future interest is in multi-objective optimization using novel biological paradigms. Only a few multi-objective PSO algorithms, such as MOPSO, and fuzzy dominance based PSO method (Koduru, Das & Welch, 2007), have been proposed; consequently, there is great interest in devising better PSO search strategies within the evolutionary computation community. Another class of algorithms based on computations involved in the vertebrate immune system is emerging, called Artificial Immune Systems (AIS). Although a few multi-objective AIS algorithms have been proposed recently (*cf.* Coello Coello & Cortés, 2005), there is substantial scope for improvement in this direction.

Other trends are in devising more difficult benchmark test problems. Huband *et al.* have proposed recent benchmarks (Huband, Hingston, Barone & While, 2006), and the performance of evolutionary methods for these functions need to be investigated.

CONCLUSION

We have provided an overview of the new and expanding field of multi-objective optimization, outlining some of the most significant approaches. We chose to describe NSGA-II and SPEA-2 as they are the most popular algorithms today. We also discuss the recent algorithm, ParEGO, which is very promising for some specialized applications as well as the even more recent FSGA, currently under development, which fills the need for hybrid multi-objective algorithms. Finally, we also have outlined MOPSO, which is based on a new evolutionary paradigm, PSO. Lastly, we address future trends in evolutionary multi-objective optimization to complete the discussion.

REFERENCES

Clerc, M., (2005). *Particle Swarm Optimization*. ISTE Press, UK.

Coello Coello, C.A. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*. 8(3): 256-279.

Coello Coello, C.A. & Cortés N.C. (2005). Solving multiobjective optimization problems using an artificial

immune system. *Genetic Programming and Evolvable Machines*. 6(2): 163-190.

Corne, D.W., Jerram, N.R., Knowles, J.D., & Oates, M.J. (2001). PESA-II: Region based selection in evolutionary multiobjective optimization, In Spector *et al.*, (editors), *Proceedings of the Genetic and Evolutionary Computation Conference*. 283-290.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley: Chichester, U.K.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182-197.

Farina M. & Amato P. (2004). A fuzzy definition of "optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics Part A-Systems and Humans*. 34(3): 315-326.

Fieldsend, J.E., Everson, R.M. & Singh, S. (2005). Multi-objective optimization in the presence of uncertainty, *IEEE Congress on Evolutionary Computation*, 1: 243-250.

Goldberg, D.E., (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.

Huband, S., Hingston, P., Barone, L., & While L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*. 10(5): 477-506.

Knowles, J.D., & Corne, D.W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*. 8: 149-172.

Knowles, J. (2005). ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, *IEEE Transactions on Evolutionary Computation*. 10(1): 50-66.

Koduru, P. Das, S., Welch, S.M., & Roe, J. (2004). Fuzzy dominance based multi-objective GA-Simplex hybrid algorithms applied to gene network models. *Proceedings of the Genetic and Evolutionary Computing Conference*, Seattle, Washington, Kalyanmoy Deb *et al.* (editors), Springer-Verlag, LNCS 3102: 356-367.

Koduru, P., Das, S., & Welch, S.M. (2007). Multi-objective and hybrid PSO using ε-fuzzy dominance,

Proceedings of the ACM Genetic and Evolutionary Computing Conference, London, UK. (Eds. Dirk Thierens et al.): 853-860.

Lu, H., & Yen, G.G. (2003). Rank-density-based multiobjective genetic algorithm and benchmark test function study. *IEEE Transactions on Evolutionary Computation*, 7(4): 325-343.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.

Zitzler, E., Laumanns, M., & Thiele, L. (2002) SPEA-2: Improving the strength Pareto approach. *Proceedings of EUROGEN 2001, Evolutionary Methods for Design, Optimization, and Control with Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty (editors), Athens, Greece: 95-100.

KEY TERMS

Elitism: A strategy in evolutionary algorithms where the best one or more solutions, called the elites, in each generation, are inserted into the next, without undergoing any change. This strategy usually speeds up the convergence of the algorithm. In a multi-objective framework, any non-dominated solution can be considered to be an elite.

Evolutionary Algorithm: A class of probabilistic algorithms that are based upon biological metaphors such as Darwinian evolution, and widely used in optimization.

Fitness: A measure that is used to determine the goodness of a solution for an optimization problem.

Fitness Landscape: A representation of the search space of an optimization problem that brings out the differences in the fitness of the solutions, such that those with good fitness are "higher". Optimal solutions are the maxima of the fitness landscape.

Generation: A term used in evolutionary algorithms that roughly corresponds to each iteration of the outermost loop. The offspring obtained in one generation become the parents of the next.

Multi-Objective Optimization: An optimization problem involving more than a single objective function.

In such a setting, it is not easy to discriminate between good and bad solutions, as a solution, which is better than another in one objective, may be poorer in another. Without any loss of generality, any optimization problem can be cast as one involving minimizations only.

Objective Function: The function that is to be optimized. In a minimization problem, the fitness varies inversely as the objective function.

Population Based Algorithm: An algorithm that maintains an entire set of candidate solutions, each solution corresponding to a unique point in the search space of the problem.

Search Space: Set of all possible solutions for any given optimization problem. Almost always, a neighborhood around each solution can also be defined in the search space.

Multi–Objective Training of Neural Networks

M. P. Cuéllar
Universidad de Granada, Spain

M. Delgado
Universidad de Granada, Spain

M. C. Pegalajar
Universidad de Granada, Spain

INTRODUCTION

Traditionally, the application of a neural network (Haykin, 1999) to solve a problem has required to follow some steps before to obtain the desired network. Some of these steps are the data preprocessing, model selection, topology optimization and then the training. It is usual to spend a large amount of computational time and human interaction to perform each task of before and, particularly, in the topology optimization and network training. There have been many proposals to reduce the effort necessary to do these tasks and to provide the experts with a robust methodology. For example, Giles et al. (1995) provides a constructive method to optimize iteratively the topology of a recurrent network. Other methods attempt to reduce the complexity of the network structure by mean of removing unnecessary network nodes and connections like in (Morse, 1994). In the last years, evolutionary algorithms have been shown as promising tools to solve this problem, existing many competitive approaches in the literature. For example, Blanco et al. (2001) proposed a master-slave genetic algorithm to train (master algorithm) and to optimize the size of the network (slave algorithm). For a general view of the problem and the use of evolutionary algorithms for neural network training and optimization, we refer the reader to (Yao, 1999).

Although the literature about genetic algorithms and neural networks is very extensive, we would like to remark the recent popularity of multi-objective optimization (Coello et al., 2002, Jin, 2006), specially to solve the problem of simultaneous training and topology optimization of neural networks. These methods have shown to perform suitably for this task in previous works, although most of them are proposed for feedforward models. They attempt to optimize the structure of the network (number of connections, hidden units or layers), while training the network at the same time. Multi-objective algorithms may provide important advantages in the simultaneous training and optimization of neural networks: They may force the search to return a set of optimal networks instead of a single one; they are capable to speed-up the optimization process; they may be preferred to a weight-aggregation procedure to cover the regularization problem in neural networks; and they are more suitable when the designer would like to combine different error measures for the training. A recent review of these techniques may be found in (Jin, 2006).

BACKGROUND

Multi-objective algorithms have become popular in the last years to solve the problem of the simultaneous training and topology optimization of neural networks, because of the innovations they can provide to solve it. Certain authors have addressed this problem through the evolution of single ensembles as for example with DIVACE-II (Chandra et al., 2006), which also implements different levels of coevolution. In other works, the networks are fully evolved and the evolutionary operators are designed to deal with both training and structure optimization. Some authors have addressed the problem of the structure optimization attending to reduce either the number of network neurons or either the number of network connections. In the first methods (Abbass et al., 2001; Delgado et al., 2005; González et al., 2003), the optimization is easier since the codification of a network contains a smaller number of freedom degrees than the last methods; however, they have a disadvantage in the sense that the networks obtained are fully connected. On the other hand, the methods in

the second place (Jin et al., 2004; Cuéllar et al, 2007) attempt to reduce the number of connections but it is not ensured that also the number of network nodes is also minimum. Nevertheless, experimental results have shown that the networks obtained with these proposals have a low size (Jin et al, 2004).

The hybridation of multi-objective evolutionary algorithms with traditional gradient-based training algorithms has also provided promising results. While the evolutionary algorithm makes a wide exploration of the solution space, the gradient-based algorithms are capable to address the search to promising areas during the evolution and to exploit the solutions suitably. This hybridation is usually carried out by including the gradient-based training method as a local search operator in the evolutionary process. Then, the local search operator is applied after the mutation and before the evaluation of the solutions. Some examples are the system MPANN developed by H.A. Abbass (2001), and the works by Y. Jin et al. (2006).

In the next section, we make an study of different aspects concerning the multi-objective optimization of neural networks. Concretely, we make an study of the objectives to be achieved in the multi-objective algorithm and the multi-objective algorithms used. We focus our analysis on recurrent neural networks (Haykin, 1999; Mandic and Chambers, 2001), since these models have a high complexity due to the recurrence. The experiments are illustrated in problems of time-series prediction, since this type of problems has multiple applications in many research and enterprise areas and the neural models used are suitable for this application, as suggested by previous works (Aussem, 1999).

MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS FOR NEURAL NETWORKS TRAINING AND OPTIMIZATION

The most recent multi-objective evolutionary algorithms are based in the concept of Pareto dominance as a criterion to determine whether a solution is optimal or not. Let $F(s)=(f_1(s),f_2(s),...,f_k(s))$ be a set of k objectives to be achieved, and let s_1 and s_2 be two solutions. In a minimization problem, it is said that s_2 is dominated by s_1 if, and only if:

$$(f_i(s_1) \leq f_i(s_2),$$
$$\forall i : 1 \leq i \leq k) \wedge (\exists j : f_j(s_1) < f_j(s_2); 1 \leq j \leq k) \quad (1)$$

The solutions that are non-nominated by any other solution are called the *non-dominated set* or *Pareto frontier*. The goal of any multi-objective algorithm is to find the solutions in the Pareto frontier. Thus, the selection of the objectives to be achieved in a multi-objective algorithm is a key aspect, since they will be used to guide the search across the search space to obtain the optimal solutions. However, the higher the number of objectives is, the higher the complexity of the search space is. In this work, we attempt to train and optimize the size of an Elman Network (Mandic and Chambers, 2001), for time series prediction problems. This network type has an input layer, an output layer and a hidden layer. The data of the time series is provided in time to the network inputs, and the objective is the network output to provide the future values of the time series at the output. The recurrent connections are in the hidden layer, so that the output of a hidden neuron at time t is also input for all the hidden neurons at time $t+1$. The reader may found a wider information about dynamical recurrent neural networks applied for time series prediction in (Aussem, 1999; Mandic and Chambers, 2001).

$$f_1(s^*) = \min\{ f_1(s)) = \min\{ \frac{1}{T}\sum_{t=1}^{T}(Y(t)-O(t))^2\} \quad (2)$$

$$f_2(s^*) = \min\{ f_2(s)\} = \min\{ h(s)\} \quad (3)$$

$$f_3(s^*) = \min\{ f_3(s)\} = \min\{ n(s)\} \quad (4)$$

For the problem of neural network optimization and training, we consider three objectives to be achieved (see equations (2)-(4)). The objective $f_1(s)$ attempts to minimize the network error, while $f_2(s)$ is used to optimize the number of hidden neurons and $f_3(s)$ the number of network connections. In equation (2), T is the number of training patterns, $Y(t)$ is the desired output for pattern t and $O(t)$ is the network output. In equation (3), $h(s)$ is the number of hidden neurons for the network s; and $n(s)$ is the number of network connections in equation (4). Another issue related to the objectives is the network codification. For example, in

M

works like in (Abbass, 2001), the objectives to achieve are $(f_1(s), f_2(s))$, obtaining fully connected networks. In this cases, the representation of the network attempts to codify the neurons if a binary vector, and the network weights in a matrix with real values. In other works like in (Jin et al., 2006), the network connections are codified into a binary matrix and the network weight into a matrix with real values, since the objectives to be optimized are $(f_1(s), f_3(s))$. If we would consider to optimize all the objectives, the representation should contain the network structure (hidden neurons and connections) and the weights. In our proposal, the number of network neurons are codified with an integer valur following the guidelines in (Delgado et al., 2005; Cuéllar et al., 2007), the connections are codified in a binary vector, and the network weights in a vector with real

values. Figure 1 shows an example of the codification of an Elman network into a solution, where V_{ij} are the network weights from input j to hidden neuron i, U_{ir} is the recurrent weight from neuron r to neuron i and W_{oi} are the weights from hidden neuron i to the output neuron o. A network connection is active if the corresponding gene is set to 1. Otherwise, the connection is not active.

The evolutionary operators like the crossover and the mutation should consider two different areas in a solution: Structural recombination/mutation, and genetic recombination/mutation. The genetic one is associated to the area of the network weights, while the structural one is for the network topology. Additionally, it could be included a local search operator to improve the network performance locally in the area that codifies the network weights, as suggested in the previous section. In our experiments, we have used a simple recombination that generates two children from two parents, with no structural recombination: We have tested that the structural recombination could provide a high exploitation of the solution space, and the selective pressure produced by the objectives to be achieved could then produce a premature convergence. On the other hand, for the mutation we have included three probabilities since the structural mutation may have a high impact in the population of solutions: Structural mutation is selected with probability p_1, and genetic mutation with probability $1 - p_1$. In the structural mutation, the number of hidden neurons is altered with probability p_2; otherwise, the active/inactive connections are mutated. Finally, a gene is altered with probability p_3. Figure 2 shows an example of the crossover, and Figure 3 shows an example of the structural mutation for the number of

Figure 1. Example for the codification of an Elman network with 1 input, 1 output, 2 hidden neurons and 5 connections
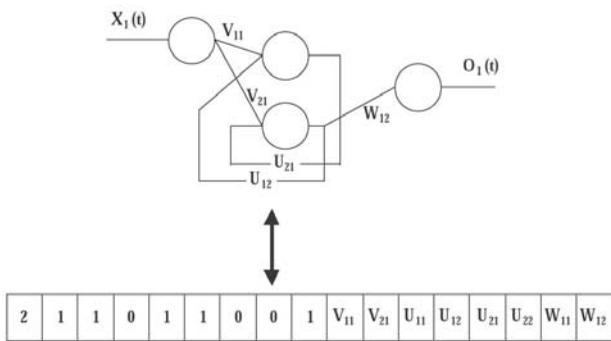
Figure 2. Example of the crossover
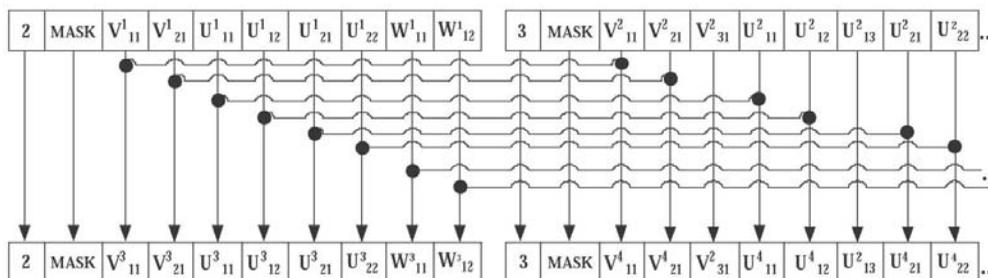
Figure 3. Example of mutation

from *http://www.economagic.com*. The parameter for the networks in our experiments are bounded for the number of hidden neurons, from 3 to 12. The networks have one input for the value of the time series at time *t*, and one output for the value of the time series at time *t+1*, to be predicted. There have been 30 experiments with the multiobjective algorithms, which are based in the algorithms *NSGA2* (Deb et al, 2002) and *SPEA2* (Zitzler et al, 2001). We label *NSGA2* and *SPEA2* for the algorithms that optimize the objectives ($f_1(s)$, $f_2(s)$) and *NSGA2.connect* and *SPEA2.connect* for the algorithms that optimize ($f_1(s)$, $f_2(s)$, $f_3(s)$). The stopping criterion is to have 10000 solutions evaluated, and size of the population is 50. The parameters for mutation are (p_1, p_2, p_3)=(0.5, 0.5, 0.1) and the range for the genes containing network weights is [-5.0, 5.0]. We have used the binary tournament selection, the heuristic Wright's crossover and the displacement mutation for the evolutionary operators.

hidden neurons. In this last case, the solution grows to have three hidden neurons, and new genes are generated. The values for these genes are a random number in the bounds of the gene. In Figure 3, these genes may be recognized by mean of the symbol *?*.

For the experiments, we attempt to study the benefits of the inclusion of a higher number of objectives to achieve in the multi-objective algorithm, and the effects of the evolutionary algorithm. To illustrate our results, we have selected two social-economic time-series for forecasting: The evolution of the U.S. population from 1950 to 2004 taken monthly (*USPop*), and the evolution of the euro/U.S. dollar variation from 1995 to 2004, taken monthly (*EurDol*). The 80% of the data are used for training, and the remaining 20% is for the test. Both time series may be downloaded for free

Figure 4 draws the distribution of the performance for the neural networks obtained in the Pareto frontiers for the 30 experiments, in each data set. Additionally, Table 1 shows the best Pareto frontiers obtained, where Column 1 plots the algorithm, columns 2 and 5 expose the number of hidden neurons, columns 3 and 6 describe the number of network connections, and columns 4 and 7 the Mean Square Error (MSE) in the training.

We may observe that *SPEA2.connect* has obtained a Pareto frontier wider than *NSGA2.connect* in both problems. In some situations, this fact may be desir-

Figure 4. Boxplots for the distribution of network performance in USPop (a), and EurDol (b)
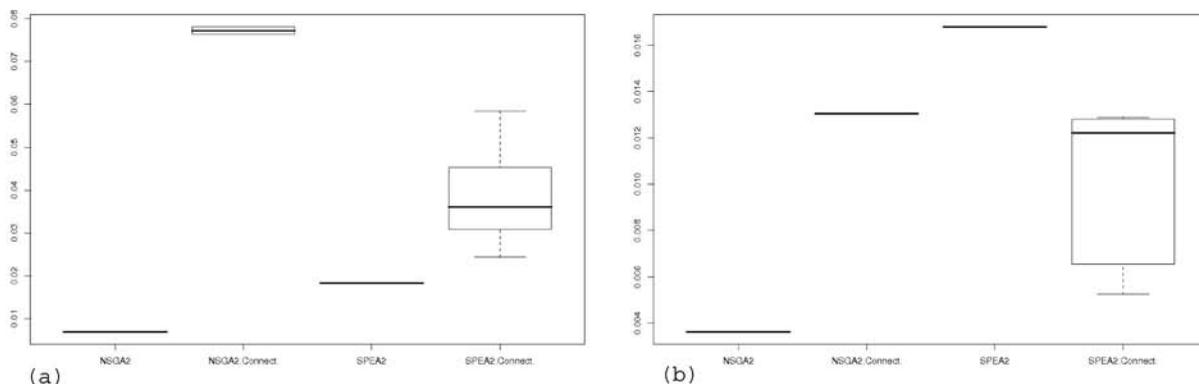
Table 1. Best Pareto frontiers obtained in the data sets

	USPop			EurDol		
Algorithm	Hidden units	Connections	MSE	Hidden units	Connections	MSE
SPEA2.connect	3	6	0.042	3	11	0.013
	5	17	0.041	4	7	0.012
	11	43	0.028	5	16	0.009
	10	44	0.050			
SPEA2	4	24	0.021	4	24	0.016
NSGA2.connect	3	10	0.092	3	4	0.013
				5	17	0.012
NSGA2	7	63	0.007	4	24	0.003

able, since we are provided with a larger set of optimal networks from we could select the best appropiate network to solve our problem. On the other hand, figures 4.a and 4.b suggest that the inclusion of the network connections optimization in the multi-objective method may produce poorer results. In both problems, the best solutions are provided by the algorithm *NSGA2*, which return fully connected networks. The same algorithm that also optimize the number of connections, *NSGA2. connect*, provides networks with minimum size, but the network performance are lower. In the case of the algorithms based on *SPEA2*, we may notice that *SPEA2.connect* is the less robust algorithm, since the distribution in the MSE is the widest. However, the boxplot shows that the best solutions of this method may be similar to the ones from *NSGA2*. This fact suggests that we may encounter smaller networks using the three objectives in *SPEA2.connect*, but sacrificing some improvements in the network performance and spending more computational time to obtain a suitable solution. Moreover, the networks obtained with the inclusion of objective $f_3(s)$ in the optimization process have a size which is very low being compared with the fully connected networks from *SPEA2* and *NSGA2*.

therefore to find optimal solutions. The hybridation of multi-objective evolutionary algorithms with non-linear programming methods to address the search space to promising areas have proved to work well in the works that propose a lower number of objectives in the optimization. In the case studied in this work, the improvements of these procedures could be better since the size of the search space is wider. Another important issue is the research of the evolution considering diversity and convergence: The objectives used usually introduce a high selective pressure in the population, and specially the objectives for the topology optimization. This could be addressed by introducing components in the evolutionary process to control the balance in diversity/convergence, therefore improving the search process and the exploration/exploitation of the solution space.

Another interesting line to work is the inclusion of objectives to improve another properties of neural networks like noise tolerance or generalization. For example, this issue has been suggested in (Graning et al, 2006), where it is introduced an extra objective to improve the generalization of a feedforward network in binary classification.

FUTURE TRENDS

We have studied in the previous section that the inclusion of a larger number of objectives for the network optimization is able to reduce the size of the network, although the network performance obtained is poorer. This is usual since the more objectives to be optimized, the more complex is the search space and

CONCLUSION

In this work, we have studied the benefits and disadvantages in multi-objective training and fully topology optimization of recurrent neural networks. We have tested the methods in time series prediction problems, and they have been also compared with the methods that do also optimize the number of connections. In general

M

terms, all the algorithms have solved the problems suitably. The methods studied provide networks with minimum number of hidden units and connections, and the network performance is

good. However, these methods may produce poorer results than those that only optimize the number of hidden neurons and provide fully connected networks. Using a higher computational time, the results from the algorithms that optimize the topology, in terms of hidden neurons and connections, may be competitive, providing networks with performance similar to those techniques that do not optimize the number of network connections. Moreover, these methods include the advantage that the network's size is very low, being compared with the fully connected networks.

REFERENCES

Abbass, H.A.; & Sarker, R. (2001). Simultaneous Evolution of Architectures and Connection Weights in ANNs. In Proc. of The Artificial neural networks and Expert systems Conference. 16-21.

Abbass, H.A. (2001). A Memetic Pareto Evolutionary Approach to Artificial Neural Networks. Lecture Notes in Artificial Intelligence. 2256, 1-12.

Aussem, A. (1999). Dynamical Recurrent Neural Networks towards Prediction and Modelling of Dynamical Systems. Neurocomputing. 28(15), 207-232.

Blanco, A.; Delgado, M. & Pegalajar, M.C. (2001). A genetic algorithm to obtain the optimal recurrent neural network. International Journal of Approximate Reasoning. 23, 67-83.

Chandra, A.; & Yao, X. (2006). Evolving hybrid ensembles of learning machines for better generalization. Neurocomputing. 69, 686-700.

Coello, C.A.; Van Veldhuizen, D.A.; & Lamont G.B. (2002). Evolutionary Algorithms for Solving Multi-objective Problems. New York: Kluwer Academic Publishers.

Cuéllar, M.P.; Delgado, M.; & Pegalajar, M.C. (2007). Topology optimization and training of Recurrent Neural Networks with pareto-based multi-objective algorithms: A experimental study. Lectures notes on Computer Science. 4507, 359-366.

Deb, K.; Patrap, A.; Agarwal, A.; & Meyarivan, T. (2002). A fase and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 6(2), 182-197

Delgado, M.; & Pegalajar, M.C. (2005). A multiobjective genetic algorithm for obtaining the optimal size of a recurrent neural network for grammatical inference. Pattern Recognition. 38(9), 1444-1456

Graning, L., Jin, Y.; & Sendhoff, B. (2006). Generalization improvement in multi-objective learning. In Proc. of International Joint Conference on Neural Networks. 9893-9900.

González, J.; Rojas, I.; Ortega, J.; Pomares, H.; Fernández, F.; & Díaz, A. (2003). Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. IEEE Transactions on Neural Networks. 14(6), 1478-1495.

Giles, C.; Chen, D.; Sun, G.; Chen, H.; Lee, H. & M. Goudreau (1995). Constructive learning of recurrent neural networks: problems with recurrent cascade correlation and a simple solution. IEEE Transactions on Neural Networks. 6(4), 489.

Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. Prentice Hall.

Jin, Y. (2006). Multi-objective machine learning. Springer, New York.

Jin, Y.; Okabe, T.; & Sendhoff B. (2004). Neural network regularization and ensembling using multi-objective evolutionary algorithms. In Proc. of the 2004 congress on evolutionary computation. 1-8.

Jin, Y.; Sendhoff, B.; & Corner, E. (2006). Evolutionary multi-objective optimization for simultaneous generation of signal-type and symbol-type representations. Lecture Notes on Computer Science. 3410, 752-766.

Mandic, D.P., & Chambers, J. (2001). Recurrent Neural Networks for Prediction. John Wiley and Sons.

Morse, J. (1994). Reducing the size of the non-dominated set, pruning by clustering. Computational and Operational Research 7(1:2), 55-66.

Yao, X. (1999). Evolving Artificial Neural Networks. Proc. Of the IEEE. 87(9), 1423-1447..

Zitzler, E.; Laummans, M.; & Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

KEY TERMS

Dynamical Recurrent Neural Networks: Artificial neural network that include recurrent connections in the network structure. They are capable to process patterns with undetermined size and/or indexed in time. The output in these networks at time *t+1* are computed using the network inputs at time *t* and the network state, provided by the recurrent connections.

Ensembles: Self-containing area of a neural network (neuron, connection, set of a neuron with connections...) that, being combined with other ensembles, is able to build a neural network that solves a problem.

Evolutionary Algorithm: Optimization algorithm based on Darwinian nature evolution.

Multi-Objective Optimization: Optimization of a problem that involves the satisfacibility or optimization of two or more objectives, sometimes opposed each other.

Regularization: Optimization of both complexity and performance of a neural network following a linear aggregation or a multi-objective algorithm.

Time-Series: Data sequence indexed in time.

Time-Series Prediction: Problem that involves the prediction of the future values of a time series, considering a few values from the data set in the past.

"Narrative" Information and the NKRL Solution

Gian Piero Zarri
LaLIC, University Paris 4-Sorbonne, France

INTRODUCTION

In a companion article of this Encyclopaedia: 'Narrative' Information, the Problem, we have introduced the problem of finding a complete and computationally efficient system for representing and managing '*nonfictional narrative information*'. We have stressed there the *important economic value of this multimedia type of information* – that concerns, e.g., corporate memory documents, news stories, normative and legal texts, medical records, intelligence messages, surveillance videos or visitor logs, actuality photos, eLearning and Cultural Heritage material, etc. We have also emphasised that the usual Computer Science tools – including those pertaining to the now very popular 'Semantic Web' domain, see (Bechhofer *et al.*, 2004, Beckett, 2004) – *are not really suitable* for dealing with this type of information.

BACKGROUND

In this article, we will present an Artificial Intelligence tool, NKRL (Narrative Knowledge Representation Language) that has been especially developed for dealing in an 'intelligent' way with the nonfictional narrative information. NKRL is, at the same time:

- a *knowledge representation system* for describing in the best possible detail the essential content (the 'meaning') of complex nonfictional 'narratives';
- a *system of reasoning (inference) procedures* that, thanks to the richness of the representation system, is able to automatically establish 'interesting' relationships among the represented data;
- an *implemented software environment* that allows the user to encode the original narratives in terms of the representation language to create 'NKRL knowledge bases' in a specific application domain and to exploit 'intelligently' these bases.

The main innovation introduced by NKRL with respect to the usual ontological paradigms concerns the addition to the traditional *ontology of concepts* – called HClass, 'hierarchy of classes' in the NKRL's jargon – an *ontology of events*, i.e., a new sort of hierarchical organization where the nodes correspond to *n*-ary structures called '*templates*' (HTemp, 'hierarchy of templates'). A partial image of the 'upper level' of HClass – that follows then the standard Protégé approach, see (Noy *et al.*, 2000) – is given in Figure 1; for HTemp, see Table 1 and Figure 2 below.

A SHORT DESCRIPTION OF NKRL

Instead of using the traditional (binary) *attribute/value* organization, the templates are generated from the

Figure 1. A partial representation of the 'upper level' of HClass, the NKRL 'traditional' ontology of concepts.
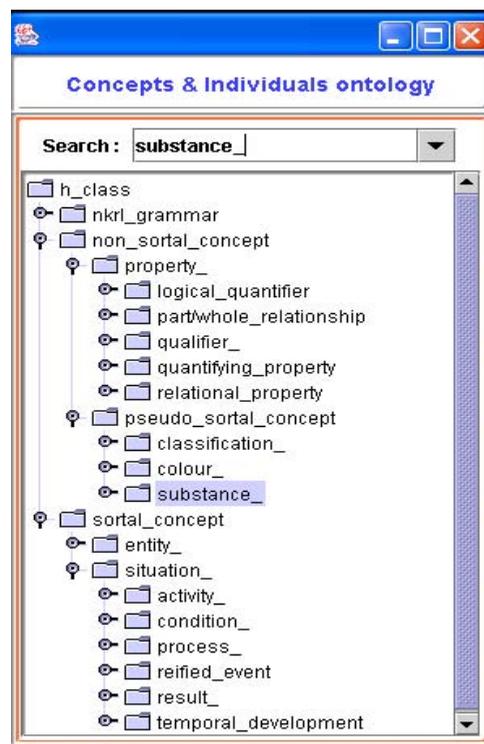
n-ary combination of quadruples connecting together the *symbolic name* of the template, a *predicate*, and the *arguments* of the predicate introduced by named relations, the *roles*. The quadruples have in common the *name* and *predicate* components. Denoting then with L_i the generic symbolic label identifying a given template, with P_j the predicate used in the template, with R_k the generic role and with a_k the corresponding argument, the core data structure for templates has the following general format (see also the companion article, 'Narrative' Information, the Problem):

$$(L_i (P_j (R_1 a_1) (R_2 a_2) \ldots (R_n a_n))) \qquad (1)$$

Predicates pertain to the set {BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE}, and roles to the set {SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), MODAL(ity), TOPIC, CONTEXT}. An argument of the predicate can consist of a simple 'concept' or of a structured association ('expansion') of several concepts. Templates can be conceived as the *formal representation* of *generic classes of elementary events* like "move a physical object", "be present in a place", "produce a service", "send/receive a message", etc. When a particular event pertaining to one of these general classes must be represented, the corresponding template is *instantiated* to produce a *predicative occurrence*.

To represent then a simple narrative like: "On November 20, 1999, in an unspecified village, an armed group of people has kidnapped Robustiniano Hablo", we must then select firstly in the HTemp hierarchy the template corresponding to "execution of violent actions", see Figure 2 and Table 1 below – this example refers to a recent application of NKRL in a 'terrorism' context in the framework of an European project see, e.g., (Zarri, 2005).

As it appears from Table 1a, the arguments of the predicate (the a_k terms in (1)) are represented by *variables with associated constraints* expressed as HClass concepts or combinations of concepts. When deriving a predicative occurrence (an instance of a template) like mod3.c5 in Table 1b, the role fillers in this occurrence must conform to the constraints of the father-template. For example, ROBUSTINIANO_HABLO (the 'BEN(e)F(iciary)' of the action of kidnapping) and INDIVIDUAL_PERSON_20 (the unknown 'SUBJECT', actor, initiator etc. of this action) are both 'individuals', instances of the HClass concept individual_person. The

constituents – as SOURCE in Table 1a – included in square brackets are optional. A 'conceptual label' like mod3.c5 is the symbolic name used to identify the NKRL code corresponding to a specific predicative occurrence.

The 'attributive operator', SPECIF(ication), is one of the four operators used in NKRL for the construction of 'structured arguments' ('complex fillers' or 'expansions') see, e.g., (Zarri, 2003). The SPECIF lists, with syntax (SPECIF e_i p_1 … p_n), are used to represent the properties or attributes that can be asserted about the first element e_i, concept or individual, of the list – e.g., in the SUBJ filler of mod3.c5, Table 1b, the attributes weapon_wearing and (SPECIF cardinality_ several_)) are both associated with INDIVIDUAL_PERSON_20.

The 'location attributes', represented in the predicative occurrences as lists, are linked with the arguments of the predicate by using the colon operator, ':', see the individual VILLAGE_1 in Table 1b. In the occurrences, the two operators date-1, date-2 materialize the temporal interval normally associated with narrative events, see (Zarri, 1998) – and, more in general, (Allen, 1981, Ferro *et al.*, 2005).

150 templates are permanently inserted into HTemp; Figure 2 reproduces the 'external' organization of the PRODUCE branch of HTemp. This branch includes the Produce:Violence template used in Table 1. HTemp corresponds then to a sort of 'catalogue' of narrative formal structures, that are very easy to 'customize' to derive the new templates that could be needed for a particular application.

What expounded until now illustrates the NKRL solutions to the problem of representing 'elementary' (simple) events. To deal now with those 'connectivity phenomena' that arise when several elementary events are connected through causality, goal, indirect speech etc. links – see also (Mani and Pustejovsky, 2004) – the basic NKRL knowledge representation tools have been complemented by more complex mechanisms that make use of second order structures, see (Zarri, 2003). For example, the *binding occurrences* consist of lists of symbolic labels (c_i) of predicative occurrences; the lists are differentiated using specific binding operators like GOAL, CONDITION and CAUSE. Let us suppose that, in Table 1, we state now that: "…an armed group of people has kidnapped Robustiniano Hablo *in order to* ask his family for a ransom", where the new elementary event: "the unknown individuals will ask for a ransom" corresponds to a new predicative occurrence, e.g., mod3.

Table 1. Building up and querying predicative occurrences

a)
name: Produce:Violence
father: Produce:PerformTask/Activity
position: 6.35
NL description: 'Execution of Violent Actions on the Filler of the BEN(e)F(iciary) Role'

      PRODUCE SUBJ   *var1*: [(*var2*)]
      OBJ   *var3*
      [SOURCE   *var4*: [(*var5*)]]
      BENF   *var6*: [(*var7*)]
      [MODAL   *var8*]
      [TOPIC   *var9*]
      [CONTEXT *var10*]
            {[modulators], ≠abs}

      var1   =   *<human_being_or_social_body>*
      var3   =   *<violence_>*
      var4   =   *<human_being_or_social_body>*
      var6   =   *<human_being_or_social_body>*
      var8   =   *<criminality/violence_related_tool>* | *<machine_tool>* | *<violence_>* |
                  <weapon_>
      var9   =   *<h_class>*
      var10   =   *<situation_>* | *<spatio/temporal_relationship>* | *<symbolic_label>*
      var2, var5, var7 = <geographical_location>

b)
mod3.c5)   PRODUCE   SUBJ   (SPECIF INDIVIDUAL_PERSON_20 weapon_wearing
                                                        (SPECIF cardinality_ several_)):
(VILLAGE_1)
                                          OBJ   kidnapping_
                                          BENF   ROBUSTINIANO_HABLO
                                          CONTEXT #mod3.c6
                              date-1:   20/11/1999
                                          date-2:
Produce:Violence (6.35)

*On November 20, 1999, in an unspecified village (*VILLAGE_1*), an armed group of people has kidnapped Robustiniano Hablo.*

c)
      PRODUCE
      SUBJ : human_being :
      OBJ : violence_
      BENF : human_being :
      {}
      date1 : 1/1/1999
      date2 : 31/12/1999

There is any information in the system concerning violence activities during 1999?

Figure 2A. Partial representation of the PRODUCE *branch of HTemp, the 'ontology of events'*
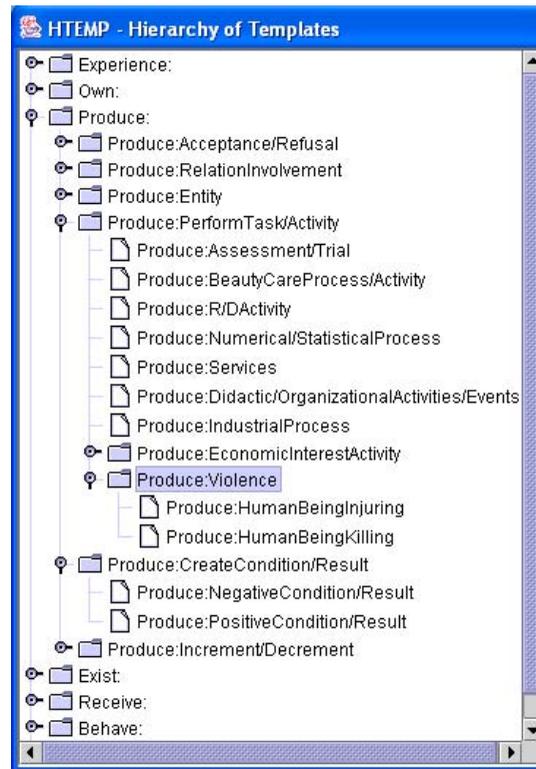
c7. To represent this situation, we must add to the occurrences that represent the two elementary events a new *binding occurrence*, e.g., mod3.c8, to link together the conceptual labels mod3.c5 (corresponding to the kidnapping occurrence, see also Table 1b) and mod3.c7 (corresponding to the new occurrence describing the intended result). mod3.c8 will have then the form: "mod3.c8) (GOAL mod3.c5 mod3.c7)". The meaning of mod3.c8 can be paraphrased as: "the activity described in mod3.c5 is focalised towards (GOAL) the realization of mod3.c7".

Reasoning in NKRL ranges from the direct questioning of an NKRL knowledge base making use of *search patterns* (formal queries over the contents of the knowledge base) that try to unify the predicative occurrences of the base to *high-level inference procedures*. A simple example of search pattern in supplied in Table 1c, producing as an answer, among other things, the predicative occurrence mod3.c5 of Table 1b – see (Ellis, 1995, Corbett, 2003, etc.) for the techniques used to unify complex conceptual structures. With respect now to the high level procedures – a detailed

paper on this topic is (Zarri, 2005) – the *transformation rules* try to 'adapt', from a semantic point of view, the original query/queries (search patterns) that failed to the real contents of the existing knowledge bases. The principle employed consists in using rules to *automatically 'transform'* the original query (i.e., the original search pattern) into one or more different queries (search patterns) that are not strictly 'equivalent' but only 'semantically close' to the original one. Let us suppose that, e.g., during the search for all the possible information linked with the Robustiniano Hablo's kidnapping, we ask the system whether Robustiano Hablo is wealthy. In the absence of a direct answer, the system will automatically 'transform' the original query using a rule like: "In a context of ransom kidnapping, the certification that a given character is wealthy or has a professional role can be substituted by the certification that: i) this character has a tight kinship link with another person, and ii) this second person is a wealthy person or a professional people". The final result can then be paraphrased in this way: we do not know whether Robustiano Hablo is wealthy, but we

can say that his father is a wealthy businessperson, see (Zarri, 2005) for the details.

Hypothesis rules allow building up *'reasonable' logic/semantic connections among the data stored in an NKRL knowledge base* using a number of pre-defined reasoning schemata, e.g., *'causal' schemata*. For example, to mention a 'classic' NKRL example, after having directly retrieved through the use of a search pattern an information like: "Pharmacopeia, an USA biotechnology company, has received 64,000,000 dollars from the German company Schering in connection with an R&D activity", we could be able to automatically construct a sort of 'causal explanation' of this event by retrieving information like: i) "Pharmacopeia and Schering have signed an agreement concerning the production by Pharmacopeia of a new compound" and ii) "in the framework of the agreement previously mentioned, Pharmacopeia has actually produced the new compound".

In Table 2, we give the informal description of the reasoning steps (called 'condition schemata' in a hypothesis context) that must be validated to prove that a generic 'kidnapping' corresponds, in reality, to a more precise 'kidnapping for ransom' environment. When several reasoning steps must be *simultaneously* validated, as in Table 2, a failure is always possible. To overcome this problem – and, at the same time, discover all the possible *implicit information* associated with the original data – the two inference modes, transformation and hypotheses, can be used in an *integrated way*, see (Zarri, 2005). In practice, we make use of 'transformations' within a 'hypothesis' context. This means that, whenever a 'search pattern' is derived from a 'condition schema' of a hypothesis to implement one of the steps of the reasoning process, we can use it 'as it is' – i.e., as originally coded when the inference rule has been built up – but also in a *'transformed' form* if the appropriate

transformation rules exist within the system.

Making use of the transformation rules already existing within the system, the hypothesis represented in an informal way in Table 2 becomes, in practice, *potentially equivalent* to the hypothesis of Table 3. For example, the proof that the kidnappers are part of a terrorist group or separatist organization (reasoning step Cond1 of Table 2) can be now obtained *indirectly*, transformation T3, by checking whether they are members of a specific subset of this group or organization.

FUTURE TRENDS

NKRL is a fully implemented language/environment. The software exists in two versions, an ORACLE-supported and a file-oriented one. Future improvements will concern mainly:

- The addition of features that will allow us querying the system in Natural Language. Very encouraging experimental results have already been obtained in this context thanks to the combined use of shallow parsing techniques – see, e.g., (Koster, 2004) and of the standard NKRL inference capabilities.
- On a more ambitious basis, the introduction of some features for the semi-automatic construction of the knowledge base of annotation/occurrences making use of full NL techniques. Some preliminary work in this context has been realised making use of the syntactic/semantic Cafetière tools, see (Black *et al.*, 2003, 2004).
- The introduction of optimisation techniques for the (basic) chronological backtracking of the NKRL InferenceEngine, in the style of the well-known techniques developed in a Logic Programming context see, e.g., (Clark and Tärnlund, 1982).

Table 2. Inference steps for the 'kidnapping for ransom' hypothesis

(Cond1) The kidnappers are part of a separatist movement or of a terrorist organization.
(Cond2) This separatist movement or terrorist organization currently practices ransom kidnapping of particular categories of people.
(Cond3) In particular, executives or assimilated categories are concerned.
(Cond4) It can be proved that the kidnapped is really a businessperson or assimilated.

Even in its present form, NKRL has been able to deal successfully, in a 'intelligent information retrieval' mode, with the most different 'narrative' domains – from history of France to terrorism, from Falkland War to the corporate domain, from the legal field to the beauty care domain or the analysis of customers' motivations, etc.

CONCLUSION

In this article, we have supplied some details about NKRL (Narrative Knowledge Representation Language), a fully implemented, up-to-date knowledge representation and inference system especially created for an 'intelligent' exploitation of narrative knowledge. The main innovation of NKRL consists in associating with the traditional ontologies of concepts an 'ontology of events', i.e., a hierarchical arrangement where the nodes correspond to *n*-ary structures called 'templates'.

REFERENCES

Allen, J.F. (1981). An Interval-Based Representation of Temporal Knowledge. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - IJCAI/81*. San Francisco: Morgan Kaufmann.

Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., and Stein, L.A., eds. (2004). *OWL Web Ontology Language Reference – W3C Recommendation 10 February 2004*. W3C (http://www.w3.org/TR/owl-ref/).

Table 3. 'Kidnapping' hypothesis in the presence of transformations concerning intermediary inference steps

(**Cond1**) The kidnappers are part of a separatist movement or of a terrorist organization.
– (**Rule T3, Consequent1**) *Try to verify whether a given separatist movement or terrorist organization is in strict control of a specific sub-group and, in this case,*
– (**Rule T3, Consequent2**) *check if the kidnappers are members of this sub-group. We will then assimilate the kidnappers to 'members' of the movement or organization.*
(**Cond2**) This movement or organization practices ransom kidnapping of given categories of people.
– (**Rule T2, Consequent**) *The family of the kidnapped has received a ransom request from the separatist movement or terrorist organization.*
– (**Rule T4, Consequent1**) *The family of the kidnapped has received a ransom request from a group or an individual person, and*
– (**Rule T4, Consequent2**) *this second group or individual person is part of the separatist movement or terrorist organization.*
– (**Rule T5, Consequent1**) *Try to verify if a particular sub-group of the separatist movement or terrorist organization exists, and*
– (**Rule T5, Consequent2**) *check whether this particular sub-group practices ransom kidnapping of particular categories of people.*
– …
(**Cond3**) In particular, executives or assimilated categories are concerned.
– (**Rule T0, Consequent1**) *In a 'ransom kidnapping' context, we can check whether the kidnapped person has a strict kinship relationship with a second person, and*
– (**Rule T0, Consequent2**) *(in the same context) check if this second person is a businessperson or*
 assimilated.
(**Cond4**) It can be proved that the kidnapped person is really an executive or assimilated.
– (**Rule T6, Consequent**) *In a 'ransom kidnapping' context, 'personalities' like physicians, journalists, artists etc. can be assimilated to businesspersons.*

N

Beckett, D., ed. (2004). RDF/XML Syntax Specification (Revised) – W3C Recommendation 10 February 2004. W3C (http://www.w3.org/TR/rdf-syntax-grammar/).

Black, W.J., McNaught, J., William J. Black, Vasilakopoulos, A., Zervanou, K., Rinaldi, F., and Theodoulidis, B. (2003). *CAFETIERE: Conceptual Annotations for Facts, Events, Individual Entities, and Relations* (Technical Report TR-U4.3.1). Manchester: UMIST Department of Computation.

Black, W.J., Jowett, S., Mavroudakis, T., McNaught, J., Theodoulidis, B., Vasilakopoulos, A., Zarri, G.P., and Zervanou, K. (2004). Ontology-Enablement of a System for Semantic Annotation of Digital Documents. In: *Proceedings of the 4th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2004) – 3rd International Semantic Web Conference* (November 8, 2004, Hiroshima, Japan).

Corbett, D. (2003). *Reasoning and Unification over Conceptual Graphs*. New York: Kluwer Academic/Plenum Publishers.

Clark, K.L., and Tärnlund, S.-A.., eds. (1982). *Logic Programming*. London: Academic Press.

Ellis, G. (1995). Compiling Conceptual Graphs. *IEEE Transactions on Knowledge and Data Engineering* **7**: 68-81.

Ferro, L., Gerber, L., Mani, I., Sundheim, B., and Wilson, G. (2005). *TIDES – 2005 Standard for the Annotation of Temporal Expressions* (2005 Release). McLean (VA): The MITRE Corporation.

Koster, C.H.A. (2004). Head/modifier Frames for Information Retrieval. In: *Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Text Processing – CLing-2004* (LNCS 2945), Gelbukh, A., ed. Berlin: Springer.

Mani, I., and Pustejovsky, J. (2004). Temporal Discourse Models for Narrative Structure. In: *Proceedings of the ACL Workshop on Discourse Annotation.* East Stroudsburg (PA): Association for Computational Linguistics.

Noy, F.N., Fergerson, R.W., and Musen, M.A. (2000). The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility. In: *Knowledge Acquisition, Modeling, and Management – Proceedings of EKAW'2000* (LNCS 1937), Dieng, R., and Corby, O., eds. Berlin: Springer.

Zarri, G.P. (1998). Representation of Temporal Knowledge in Events: The Formalism, and Its Potential for Legal Narratives. *Information & Communications Technology Law – Special Issue on Models of Time, Action, and Situations* **7**: 213-241.

Zarri, G.P. (2003). A Conceptual Model for Representing Narratives. In: *Innovations in Knowledge Engineering*, Jain, R., Abraham, A., Faucher, C., and van der Zwaag, eds. Adelaide: Advanced Knowledge International.

Zarri, G.P. (2005). Integrating the Two Main Inference Modes of NKRL, Transformations and Hypotheses. *Journal on Data Semantics (JoDS)* **4**: 304-340.

KEY TERMS

Attributive Operator: The 'attributive operator', SPECIF(ication), is one of the four operators used in NKRL for the construction of 'structured arguments' ('complex fillers' or 'expansions') of the conceptual predicates. The SPECIF lists, with syntax (SPECIF e_i p_1 … p_n), are used to represent the properties or attributes that can be asserted about the first element e_i, concept or individual, of the list.

Binding Occurrences: Second order structures used to deal with those 'connectivity phenomena' that arise when several elementary events are connected through causality, goal, indirect speech etc. links. They consists of lists of symbolic labels (c_i) of predicative occurrences; the lists are differentiated using specific binding operators like GOAL, CONDITION and CAUSE.

Format of NKRL Templates: Templates take the form of *n*-ary combinations of quadruples connecting together the 'symbolic name' of the template, a 'conceptual predicate' and the 'arguments' of the predicate introduced by named relations, the 'roles' (like SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), etc.). The quadruples have in common the 'name' and 'predicate' components. Denoting then with L_i the symbolic label identifying the template, with P_j the predicate, with R_k the generic

role and with a_k the generic argument, the core data structure for templates has the format:

$$(L_i \ (P_j \ (R_1 \ a_1) \ (R_2 \ a_2) \ \ldots \ (R_n \ a_n))) \ .$$

Templates are included in an inheritance hierarchy, HTemp(lates), which implements NKRL's 'ontology of events'.

NKRL Inference Engine: A software module that carries out the different 'reasoning steps' included in hypotheses or transformations. It allows us to use these two classes of inference rules also in an 'integrated' mode, augmenting then the possibility of finding interesting (implicit) information.

NKRL Inference Rules, Hypotheses: They are used to build up automatically 'reasonable' connections among the information stored in an NKRL knowledge base according to a number of pre-defined reasoning schemata, e.g., 'causal' schemata'.

NKRL Inference Rules, Transformations: These rules try to 'adapt', from a semantic point of view, a query that failed to the contents of the existing knowledge bases. The principle employed consists in using rules to automatically 'transform' the original query into one or more different queries that are not strictly 'equivalent' but only 'semantically close' to the original one.

Ontology of Concepts vs. Ontology of Events: The ontologies of concepts concern the 'standard' hierarchical organizations of concepts to be used to model (in a 'static' way) a given domain. NKRL adds an 'ontology of events', i.e., a new sort of hierarchical organization where the nodes, represented by *n*-ary structures called 'templates', represent general classes of 'dynamical' events like "move a physical object", "produce a service", "send/receive a message", etc.

"Narrative" Information Problems

Gian Piero Zarri
LaLIC, University Paris 4-Sorbonne, France

INTRODUCTION

'Narrative' information concerns in general the account of some real-life or fictional story (a 'narrative') involving concrete or imaginary 'personages'. In this article we deal with *(multimedia) nonfictional narratives of an economic interest*. This means, first, that we are not concerned with all sorts of *fictional* narratives that have mainly an entertainment value, and represent an imaginary narrator's account of a story that happened in an imaginary world: a novel is a typical example of fictional narrative. Secondly, our 'nonfictional narratives' must have an *economic value*: they are then typically embodied into corporate memory documents, they concern news stories, normative and legal texts, medical records, intelligence messages, surveillance videos or visitor logs, actuality photos and video fragments for newspapers and magazines, eLearning and multimedia Cultural Heritage material, etc.

Because of the ubiquity of these *'narrative', 'dynamic' resources*, it is particularly important to build up *computer-based applications* able to represent and to exploit in a general, accurate, and effective way the *semantic content* – i.e., the key *'meaning'* – of these resources.

BACKGROUND

'Narratives' represent presently a very 'hot' domain. From a *theoretical point of view*, they constitute the object of a full discipline, the 'narratology', whose aim can be defined as that of producing *an in-depth description of the 'syntactic/semantic structures' of the narratives,* i.e., the narratologist is in charge of dissecting narratives into their component parts in order to establish their functions, their purposes and the relationships among them. A good introduction to the full domain is (Jahn, 2005).

Even if narratology is particularly concerned with *literary analysis* (and, therefore, with 'fictional' narra-

tives), these last years some of its varieties have acquired a particular importance also from a strict Artificial Intelligence (AI) and Computer Science (CS) point of view. Leaving apart the old dream of generating fictions by computer, see (Mehan, 1977) and, more recently, (Callaway and Lester, 2002), we can mention here two new disciplines, 'storytelling' and 'eChronicles', that are of interest from both a nonfictional narratives and a AI/CS point of view.

Storytelling – see, e.g., (Soulier, 2006) – concerns in general the study of the different ways of *conveying 'stories' and events in words, images and sounds* in order to entertain, teach, explain etc. *Digital Storytelling* deals in particular with the ways of introducing characters and emotions in the *interactive entertainment domain*, and concerns then videogames, massively multiplayer online games, interactive TV, virtual reality etc., see (Handler Miller, 2004). Digital Storytelling is, therefore, related to another, computer-based variant of narratology called *Narrative Intelligence*, a sub-domain of AI that *explores topics at the intersection of Artificial Intelligence, media studies, and human computer interaction design* (narrative interfaces, history databases management systems, artificial agents with narrative structured behaviour, systems for the generation and/or understanding of histories/narratives etc.), see (Mateas and Sengers, 2003).

An *eChronicle system* can be defined in short as way of *recording, organizing and then accessing streams of multimedia events captured by individuals, groups, or organizations making use of video, audio and other sensors*. The 'chronicles' gathered in this way may concern any sort of 'narratives' from meeting minutes to football games, sales activities, 'lifelogs' obtained from wearable sensors, etc. The technical challenges concern mainly the ways of aggregating the events into coherent 'episodes' making use of domain models as ontologies, and providing then access to this sort of material to the users at the required level of granularity. Note that *exploration, and not 'normal' querying, is the predominant way of interaction with the chronicle*

repositories; more details can be found, e.g., in (Güven, Podlaseck and Pingali, 2005), (Westermann and Jain, 2006).

The solution (NKRL) proposed for the 'intelligent' management of nonfictional narratives in the companion article – 'Narrative' Information, the NKRL Solution – of the present one is considered as a fully-fledged eChronicle technique, see (Zarri, 2006). In NKRL, however, a fundamental aspect concerns the presence of powerful *'reasoning' techniques* – an aspect that is not taken into consideration sufficiently in depth in eChronicles that are mainly interested in the accumulation of narrative materials more than in the 'intelligent' exploitation of their inner relationships.

REPRESENTING THE 'NONFICTIONAL' NARRATIVES

All the different sorts of 'nonfictional narratives' evoked in the previous Sections concern, practically, the *description of spatially and temporally characterised 'events' that relate, at some level of abstraction, the behaviour or the state of some real-life 'actors'* (characters, personages, etc.): these try to attain a specific result, experience particular situations, manipulate some (concrete or abstract) materials, send or receive messages, buy, sell, deliver etc. Note that:

- The term 'event' is taken here in its *most general meaning*, covering also strictly related notions like fact, action, state, situation, episode, activity etc.
- The 'actors' or 'personages' involved in the events *are not necessarily human beings*: we can have narratives concerning, e.g., the vicissitudes in the journey of a nuclear submarine (the 'actor', 'subject' or 'personage') or the various avatars in the life of a commercial product.
- Even if a large amount of nonfictional narratives are embodied within natural language (NL) texts, this is *not necessarily true*: narrative information is really '*multimedia*'. A photo representing a situation that, verbalized, could be expressed as "The US President is addressing the Congress" is not of course an NL document, yet it surely represents a narrative.

An in-depth analysis of the existing Knowledge Representation solutions that *could* be used to represent and manage nonfictional narratives endowed with the above characteristics is beyond the possibilities of this article – see in this context, e.g., (Zarri, 2005). We will limit ourselves, here, to some quick consideration.

We can note, first of all, that the now so popular Semantic Web (W3C) languages like RDF (Resource Description Framework), see (Manola and Miller, 2004), and OWL (Web Ontology Language), see (McGuinness and Harmelen, 2004) are unable to fit the bill because their core formalism consists in practice of the classical '*attribute – value*' model. For these 'binary' languages then, a property can only be a *binary relationship*, linking two individuals or an individual and a value. When these languages must represent simple 'narratives' like "John has given a book to Mary", several difficulties arise. In this extremely simple sentence, e.g., "give" is an *n-ary (ternary) relationship* that, to be represented in a complete way, asks for the presence of a specific '*semantic predicate*' in the "give" or "transfer" style, where the '*arguments*', "John", "book" and "Mary", of the predicate must be labelled with '*conceptual roles*' such as, e.g., 'agent of give', 'object of give' and 'beneficiary of give' respectively.

Efforts for extending the W3C languages by introducing some *n*-ary feature have been not very successful until now: see, in this context, a recent working paper from the W3C Semantic Web Best Practices and Deployment Working Group (SWBPD WG) about "Defining N-ary Relations on the Semantic Web" (Noy and Rector, 2006). This paper proposes some extensions to the binary paradigm to allow the correct representation of 'narratives' like: "Steve has temperature, which is high, but failing" or "United Airlines flight 3177 visits the following airports: LAX, DFW, and JFK". The technical solutions expounded in this paper are not very convincing and have aroused several criticisms. These have focused, mainly, on i) the fact that the majority of the solutions proposed *do not deal, in reality, with the* n-*ary problem*, but with (only loosely) related matters like the possibility of specifying a *'standard' binary relationship* via the addition of properties, and ii) on the *arbitrary introduction, through reification processes, of fictitious (and inevitably ad hoc) 'individuals' to represent the* n-*ary relations* when these are actually dealt with. Moreover, the paper say nothing,

e.g., about the way of dealing, in concrete 'narrative' situations, with those crucial *'connectivity phenomena'* like causality, goal, indirect speech, co-ordination and subordination etc. that *link together the basic pieces of information* – e.g., the 'basic events' corresponding to the present illness state of Steve with other 'basic events' corresponding to the (possible or definite) 'causes' of such state.

Several solutions for representing narratives in computer-usable ways according to some sort of *actual 'n*-ary model' have been described in the literature. For example, in the context of his work – between the mid-fifties and the mid-sixties – on the set up of a mechanical translation process based on the simulation of the thought processes of the translator, Silvio Ceccato (Ceccato, 1961) proposed a representation of narrative-like sentences as a network of *triadic structures* ('correlations') organized around specific 'correlators' (a sort of roles). Ceccato is also credited to be one of the pioneers of the semantic network studies; basically, semantic networks are directed graphs (digraphs) where the nodes represent concepts, and the arcs different kinds of associative links, not only the 'classical' IsA and property-value links, but also *n*-ary relationships. A panorama of the different conceptual solutions proposed in a semantic network context can be found in (Lehmann, 1992).

In the seventies, a sort of particularly popular, *n*-ary semantic network approach has been represented by the Conceptual Dependency theory of Roger Schank (Schank, 1973). In this theory, the underlying meaning ('conceptualization') of narrative-like utterances is expressed as *combinations of 'semantic predicates'* chosen from a set of twelve 'primitive actions' (like INGEST, MOVE, ATRANS, the transfer of an abstract relationship like possession, ownership and control, PTRANS, physical transfer, etc.) plus states and changes of states, and *seven role relationships ('conceptual case')*. Conceptual Graphs (CGs) is the representation system developed by John Sowa (Sowa, 1984, 1999) and derived, at least partly, from Schank's work and other early work in the Semantic Networks domain. CGs make use of a *graph-based notation* for representing 'concept-types' (organized into a type-hierarchy), 'concepts' (that are instantiations of concept types) and 'conceptual relations' that relate one concept to another. CGs can be used to represents in a formal way narratives like "A pretty lady is dancing gracefully" and more complex,

second-order constructions like contexts, wishes and beliefs. CYC, see (Lenat *et al.*, 1990) concerns one of the most controversial endeavours in the history of Artificial Intelligence. Started in the early '80 as a MCC (Microelectronics and Computer Technology Corporation, Texas, USA) project, it ended about 15 years later with the set up of an *enormous knowledge base* containing about a million of hand-entered 'logical assertions' including both simple statements of facts and rules about what conclusions can be inferred if certain statements of facts are satisfied. The 'upper level' of the ontology that structures the CYC knowledge base is now freely accessible on the Web, see http://www.cyc.com/cyc/opencyc. A detailed analysis of the origins, developments and motivations of CYC can be found in (Bertino *et al.*, 2001: 275-316). We can also mention here another 'modern' system, Topic Maps, see (Rath, 2003), where information is represented using topics (representing any concept, from people to software modules and events), associations (the relationships between them), and occurrences (the relationships between topics and information resources relevant to them). They correspond, eventually, to a sort of downgraded Semantic Network representation.

Leaving now aside 'historical' solutions like those proposed by Schank or Ceccato, *none of the existing n-ary solutions mentioned above seem to be able to satisfy completely the nonfictional narratives requirements*, see again (Zarri, 2005) for more details. The universal purposes of CYC, the extremely large dimensions of its knowledge base and the extreme diversity of the contents of this base give rise to *serious consistency problems*, that have apparently restricted the development of concrete applications based on this technology to experimental projects mainly supported by the US Government. On the other hand, the knowledge representation language of CYC, CycL (substantially, a frame system rewritten in logical form) seems to be *too rigid and uniform* to adapt itself to the representation of all the different facets (from general concepts and elementary events to the connectivity phenomena etc.) that characterise the narratives. Conceptual Graphs (CGs) could represent, at least in principle, a valid solution for dealing with nonfictional narrative information. However, it seems evident that work in a CGs context concerns mainly, with few exceptions, the 'academic' domain, and that the practically-oriented applications of CGs are particularly scarce. This becomes particular

evident when we consider that the CGs developers *still lack of an exhaustive and authoritative list of standard CGs structures under the form of 'canonical graphs'* that could constitute a sort of 'catalogue' for dealing with practical problems; the set up of a tool like this seems never have been planned. The existence of such a catalogue could be extremely important for practical applications in the narrative (not only) domain given that: i) a system-builder should not have to create himself the structural and inferential knowledge needed to describe and exploit the events proper to a (sufficiently) large class of narratives; ii) the reproduction and the sharing of previous results could become neatly easier.

We can add to the above difficulties the existence of a series of *general problems* that are not associated with a specific system but that concern by and large all the existing *n*-ary solutions, like the lack of agreement about the list of 'roles' (conceptual cases) to be used when a narrative must be practically represented into conceptual format, or the differences of opinion about the use of 'primitives'.

ACTUAL TRENDS

In spite of the quite pessimistic considerations of the previous Section, conceiving a *specific Knowledge Representation tool* for dealing in practice with nonfictional narrative information is far from being impossible. Returning now to the "John gave a book…" example above – and leaving aside, for the moment being, all the additional problems linked, e.g., with the existence of the 'connectivity phenomena' – it is not too difficult to see that a complete, *n*-ary representation that captures all the '*essential meaning*' of this elementary narrative amounts to:

- Define JOHN_, MARY_ and BOOK_1 as 'individuals', instances of general 'concepts' like human_being and information_support or of more specific concepts. Concepts and instances (individuals) are, as usual, collected into a 'binary' ontology (built up using a standard tool like, e.g., Protégé).
- Define an *n*-ary structure organised around a *conceptual predicate* like, e.g., MOVE or PHYSICAL_TRANSFER and associate the above individuals (the arguments) to the predicate through the use of *conceptual roles* that specify their

'function' within the global narrative. JOHN_ will then be introduced by an AGENT (or SUBJECT) role, BOOK_1 by an OBJECT (or PATIENT) role, MARY_ by a BENEFICIARY role. An additional information like "yesterday" could be introduced by, e.g., a TEMPORAL_ANCHOR role, etc.
- 'Reify' the obtained *n*-ary structured associating with it an *unique identifier* under the form of a 'semantic label', to assure both i) the logical-semantic coherence of the structure; ii) an rational and efficient way of storing and retrieving it.

Formally, an *n*-ary structure defined according the above guidelines can be described as:

$$(L_i (P_j (R_1 a_1) (R_2 a_2) … (R_n a_n))) \tag{1}$$

where L_i is the symbolic label identifying the particular *n*-ary structure (e.g., the global structure corresponding to the representation of the "John gave a book…" example), P_j is the conceptual predicate, R_k is the generic role and a_k the corresponding argument (e.g., the individuals john_, mary_ etc.). Note that each of the (R_i a_i) cells of (1), *taken individually*, represents a binary relationship in the W3C language style. The main point here is, however, that the whole conceptual structure represented by (1) must be considered *globally*.

The solution represented formally by (1) is at the core of a complete and running conceptual tools for the representation and management of nonfictional narrative information called NKRL (Narrative Knowledge representation Language), see (Zarri, 2005) and the companion article: 'Narrative' Information, the NKRL Solution.

CONCLUSION

We deal in this article with 'nonfictional narratives'. These are information resources of a high economical importance that concern, e.g., the 'corporate knowledge' documents, the news stories, the medical records, the surveillance videos or visitor logs, etc. When we examine the existing (or past) *general* Knowledge Representation systems that could be used for dealing with nonfictional narratives, we can note that none of them seem to be able to satisfy completely the nonfictional narratives requirements. For example, the

W3C (Semantic Web) languages like RDF and OWL cannot fit the bill since they are binary-based types of representation while narratives ask, in general, for *n*-ary solutions. A *specific, narrative-oriented* formalism able to capture the essential 'meaning' of an 'elementary' narrative event however exists, see (Zarri, 2005) and the companion article: 'Narrative' Information, the NKRL Solution.

REFERENCES

Bertino, E., Catania, B., and Zarri, G.P. (2001). *Intelligent Database Systems*. London: Addison-Wesley and ACM Press.

Callaway, C.B., and Lester, J.C. (2002). Narrative Prose Generation. *Artificial Intelligence* **139**: 213-252.

Ceccato, S. (1961) *Linguistic Analysis and Programming for Mechanical Translation*. Milano: Feltrinelli.

Güven, S., Podlaseck, M., and Pingali, G. (2005). PICASSO: Pervasive Information Chronicling, Access, Search, and Sharing for Organizations. In: *Proceedings of the IEEE 2005 Pervasive Computing Conference (PerCom 2005)*. Los Alamitos (CA): IEEE Computer Society Press.

Handler Miller, C. (2004). *Digital Storytelling. A Creator's Guide to Interactive Entertainment*. Burlington (MA): Focal Press.

Jahn, M. (2005). *Narratology: A Guide to the Theory of Narrative* (version 1.8). Cologne: English Department of the University (http://www.uni-koeln.de/~ame02/pppn.htm).

Lehmann, F., ed. (1992). *Semantic Networks in Artificial Intelligence*. Oxford: Pergamon Press.

Lenat, D.B., Guha, R.V., Pittman, K., Pratt, D., and Shepherd, M. (1990). CYC: Toward Programs With Common Sense. *Communications of the ACM* **33**(8): 30-49.

Manola, F., and Miller, E. (2004). *RDF Primer – W3C Recommendation 10 February 2004*. W3C (http://www.w3.org/TR/rdf-primer/).

Mateas, M., and Sengers, P., eds. (2003). *Narrative Intelligence*. Amsterdam: John Benjamins.

McGuinness, D.L., van Harmelen, F. (2004). *OWL WEB Ontology Language Overview – W3C Recommendation 10 February 2004*. W3C (http://www.w3.org/TR/owl-features/).

Mehan, J. (1977). TALE-SPIN – An Interactive Program That Writes Stories. In: *Proceedings of the 1977 International Joint Conference on Artificial Intelligence - IJCAI/97*. San Mateo (CA): Morgan Kaufmann.

Rath, H.H. (2003). *The Topic Maps Handbook* (White Paper, version 1.1). Gütersloh: empolis GmbH (http://www.empolis.com/downloads/empolis_TopicMaps_Whitepaper20030206.pdf).

Noy, F.N., and Rector, A., eds. (2006). *Defining N-ary Relations on the Semantic Web – W3C Working Group Note 12 April 2006*. W3C (http://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412/).

Schank, R.C. (1973). Identification of Conceptualizations Underlying Natural Language. In: *Computer Models of Thought and Language*, Schank, R.C., and Colby, K.M., eds. San Francisco: W.H. Freeman and Co.

Soulier, E., ed. (2006). *Le Storytelling, concepts, outils et applications*. Paris : Lavoisier.

Sowa, J.F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Reading (MA): Addison-Wesley.

Sowa, J.F. (1999). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove (CA): Brooks Cole Publishing Co.

Westermann, U., and Jain, R. (2006). A Generic Event Model for Event-Centric Multimedia Data Management in eChronicle Applications. In: *Proceedings of the 22nd International Conference on Data Engineering Workshops – ICDE Workshop on eChronicles (ICDEW'06)*. Los Alamitos (CA): IEEE Computer Society Press.

Zarri, G.P. (2005). Integrating the Two Main Inference Modes of NKRL, Transformations and Hypotheses. *Journal on Data Semantics (JoDS)* **4**: 304-340.

Zarri, G.P. (2006). Modeling and Advanced Exploitation of eChronicle 'Narrative' Information. In: *Proceedings of the 22nd International Conference on Data Engineering Workshops – ICDE Workshop on eChronicles (ICDEW'06)*. Los Alamitos (CA): IEEE Computer Society Press.

KEY TERMS

'Binary' Languages vs. *n*-ary Languages: Binary languages (like RDF and OWL) are based on the classical 'attribute – value' model: they are called 'binary' because, for them, a property can only be a binary relationship, linking two individuals or an individual and a value. They cannot be used to represent in an accurate way the narratives that ask in general, on the contrary, for the use of *n*-ary knowledge representation languages.

Connectivity Phenomena: In the presence of several, logically linked elementary events, this term denotes the existence of a global 'narrative' information content that goes beyond the simple addition of the information conveyed by the single events. The connectivity phenomena are linked with the presence of logico-semantic relationships like causality, goal, co-ordination and subordination etc.

Core Format of a Complete Solution for Representing Narratives: Formally, an *n*-ary structure able to represent the 'essential meaning' of an 'elementary event' can be described as:

$$(L_i \, (P_j \, (R_1 \, a_1) \, (R_2 \, a_2) \, \ldots \, (R_n \, a_n)))$$

where L_i is the symbolic label identifying the particular formalized event, P_j is the conceptual predicate, R_k is the generic role and a_k the corresponding argument.

Examples of *n*-ary Languages: 'Historical' examples of *n*-ary languages are Ceccato's 'correlations', Schank's Conceptual Dependency theory, many Semantic Networks proposals, etc. Current *n*-ary systems are, e.g., Topic Maps, Sowa's Conceptual Graphs, Lenat's CYC, etc. None of them are able to satisfy completely the requirements for an 'intelligent' representation and management of nonfictional narrative information.

Narrative Information: Concerns in general the account of some real-life or fictional story (a 'narrative') involving concrete or imaginary 'personages'.

Narratology: Discipline that deals with narratives from a theoretical point of view. Sub-classes of narratology that have a 'computational' interest are, e.g., Storytelling, Narrative Intelligence and the eChronicle systems.

Nonfictional Narrative of an Economic Interest: In this case, the personages are 'real characters', and the narrative happens in the real world. Moreover, the narratives are now embodied in multimedia documents of an economic interest: corporate memory documents, news stories, normative and legal texts, medical records, intelligence messages, surveillance videos or visitor logs, etc.

Natural Language Processing and Biological Methods

Gemma Bel Enguix
Rovira i Virgili University, Spain

M. Dolores Jiménez López
Rovira i Virgili University, Spain

INTRODUCTION

During the 20th century, biology—especially molecular biology—has become a pilot science, so that many disciplines have formulated their theories under models taken from biology. Computer science has become almost a bio-inspired field thanks to the great development of natural computing and DNA computing.

From linguistics, interactions with biology have not been frequent during the 20th century. Nevertheless, because of the "linguistic" consideration of the genetic code, molecular biology has taken several models from formal language theory in order to explain the structure and working of DNA. Such attempts have been focused in the design of grammar-based approaches to define a combinatorics in protein and DNA sequences (Searls, 1993). Also linguistics of natural language has made some contributions in this field by means of Collado (1989), who applied generativist approaches to the analysis of the genetic code.

On the other hand, and only from theoretical interest a strictly, several attempts of establishing structural parallelisms between DNA sequences and verbal language have been performed (Jakobson, 1973, Marcus, 1998, Ji, 2002). However, there is a lack of theory on the attempt of explaining the structure of human language from the results of the semiosis of the genetic code. And this is probably the only arrow that remains incomplete in order to close the path between computer science, molecular biology, biosemiotics and linguistics.

Natural Language Processing (NLP) –a subfield of Artificial Intelligence that concerns the automated generation and understanding of natural languages— can take great advantage of the structural and "semantic" similarities between those codes. Specifically, taking the systemic code units and methods of combination of the genetic code, the methods of such entity can be translated to the study of natural language. Therefore, NLP could become another "bio-inspired" science, by

means of theoretical computer science, that provides the theoretical tools and formalizations which are necessary for approaching such exchange of methodology.

In this way, we obtain a theoretical framework where biology, NLP and computer science exchange methods and interact, thanks to the semiotic parallelism between the genetic code and natural language.

BACKGROUND

Most current natural language approaches show several facts that somehow invite to the search of new formalisms to account in a simpler and more natural way for natural languages. Two main facts lead us to look for a more natural computational system to give a formal account of natural languages: a) natural language sentences cannot be placed in any of the families of the Chomsky hierarchy (Chomsky, 1956) in which current computational models are basically based, and b) rewriting methods used in a large number of natural language approaches seem to be not very adequate, from a cognitive perspective, to account for the processing of language.

Now, if to these we add (1) that languages that have been generated following a molecular computational model are placed in-between Context-Sensitive and Context-Free families; (2) that genetic model offers simpler alternatives to the rewriting rules; (3) and that genetics is a natural informational system as natural language is, we have the ideal scene to propose biological models in NLP.

The idea of using biological methods in the description and processing of natural languages is backed up by a long tradition of interchanging methods in biology and natural/formal language theory:

1. **Results and methods in the field of formal language theory have been applied to biology**:

(1) Pawlak (1965) dependency grammars as an approach in the study of protein formation; (2) transformational grammars for modeling gene regulations (Collado, 1989); (3) stochastic context-free grammars for modeling RNA (Sakakibara et al., 1994); (4) definite clause grammars and cut grammars to investigate gene structure and mutations and rearrangement in it (Searls, 1989); (5) tree-adjoining grammars for predicting RNA structure of biological data (Uemura et al., 1999).

2. **Natural languages as models for biology:** (1) Watson (1968) understanding of heredity as a form of communication; (2) Asimov (1968) idea that nucleotide bases are letters and they form an alphabet; (3) Jacob (1970) consideration that the sense of the genetic message is given by the combination of its signs in words and by the arrangement of words in phrases; (4) Jakobson (1970) ideas about taking the nucleotide bases as phonemes of the genetic code or about the binary oppositions in phonemes and in the nucleic code.

3. **Biological ideas in linguistics:** (1) the "tree model" proposed by Schleicher (1863); (2) the "wave model" due to Schmidt (1872); (3) the "geometric network model" proposed by Forster (1997); or (3) the naturalistic metaphor in Linguistics defended by Jakobson (1970, 1973).

4. **Using DNA as a support for computation** is the basic idea of Molecular Computing (Păun et al., 1998). Speculations about this possibility can be found in Feynman (1961), Bennett (1973) and Conrad (1995).

BIOLOGICAL METHODS IN NLP

Here, we present an overview of different bio-inspired methods that during the last years have been successfully applied to several NLP issues, from syntax to pragmatics. Those methods are taken mainly from computer science and are basically the following: *DNA computing, membrane computing and networks of evolutionary processors*.

DNA Computing

One of the most developed lines of research in natural computing is the named molecular computing, a model based on molecular biology, which arose mainly after Adleman (1994). An active area in molecular computing is DNA computing (Păun et al., 1998) inspired in the way that DNA perform operations to generate, replicate or change the configuration of the strings.

Application of molecular computing methods to natural language syntax gives rise to **molecular syntax** (Bel-Enguix & Jiménez-López, 2005a). Molecular syntax takes as a model two types of mechanisms used in biology in order to modify or generate DNA sequences: *mutations* and *splicing*. Mutations refer to changes performed in a linguistic string, being this a phrase, sentence or text. Splicing is a process carried out involving two or more linguistic sequences. It is a good framework for approaching syntax, both from the sentential or dialogical perspective.

Methods used by molecular syntax are based on basic genetic processes: *cut*, *paste*, *delete* and *move*. Combining these elementary rules most of the complex structures of natural language can be obtained, with a high degree of s*implicity*.

This approach is a test of the generative power of splicing for syntax. It seems, according to the results achieved, that splicing is quite powerful for generating, in a very simple way, most of the patterns of the traditional syntax. Moreover, the new perspectives and results it provides, could mean a transformation in the general perspective of syntax.

From here, we think that bio-NLP, applied in a methodological and clear way, is a powerful and simple model that can be very useful to a) formulate some systems capable of generating the larger part of structures of language, and b) define a formalization that can be implemented and may be able to describe and predict the behavior of natural language structures.

Membrane Computing

Membrane Systems (MS) (Păun, 2000) are models of computation inspired by some basic features of biological membranes. They can be viewed as a new paradigm in the field of natural computing based on the functioning of membranes inside the cell. MS can be used as generative, computing or decidability devices. This new computing model has several intrinsically

interesting features such as, for example, the use of multisets and the inherent parallelism in its evolution and the possibility of devising computations which can solve exponential problems in polynomial time.

This framework provides a powerful tool for formalizing any kind of interaction, both among agents and among agents and environment. One of key ideas of MS is that generation is made by evolution. Therefore, most of evolving systems can be formalized by means of membrane systems.

Linguistic Membrane Systems (LMS) (Bel-Enguix & Jiménez-López, 2005b) aim to model linguistic processes, taking advantage of the flexibility of MS and their suitability for dealing with some fields where contexts are a central part of the theory. LMS can be easily adapted to deal with different aspects of the description and processing of natural languages. The most developed applications of LMS are *semantics* and *dialogue*.

MS are a good framework for developing a semantic theory because they are evolving systems by definition, in the same sense that we take meaning to be a dynamic entity. Moreover, MS provide a model in which contexts, either isolated or interacting, are an important element which is already formalized and can give us the theoretical tools we need. Semantic membranes may be seen as an integrative approach to semantics coming from formal languages, biology and linguistics. Taking into account results obtained in the field of computer science as well as the naturalness and simplicity of the formalism, it seems the formalization of contexts by means of membranes is a promising area of research for the future. Examples of application of MS to semantics can be found in Bel-Enguix and Jiménez-López (2007).

A topic where context and interaction among agents is essential is the field of dialogue modeling and its applications to the design of effective and user-friendly computer dialogue systems. Taking into account a pragmatic perspective of dialogue and based on speech acts, multi-agent theory and dialogue games, Dialogue Membrane Systems have arisen, as an attempt to compute speech acts by means of MS. Considering membranes as agents, and domains as a personal background and linguistic competence, the application to dialogue is almost natural, and simple from the formal point of view. For examples of this application see Bel-Enguix and Jiménez-López (2006b).

NEPS-Networks of Evolutionary Processors

Networks of Evolutionary Processors (NEPs) are a new computing mechanism directly inspired in the behavior of cell populations. Every cell is described by a set of words (DNA) evolving by mutations, which are represented by operations on these words. At the end of the process, only the cells with correct strings will survive. In spite of the biological inspiration, the architecture of the system is directly related to the Connection Machine (Hillis, 1985) and the Logic Flow paradigm (Errico et al. 1994). Moreover, the global framework for the development of NEPs has to be completed with the biological background of DNA computing (Păun et al., 1998), membrane computing (Păun, 2000) and, specially, with grammar systems (Csuhaj-Varjú et. al., 1994), which share with NEPs the idea of several devices working together and exchanging results.

First precedents of NEPs as generating devices can be found in Csuhaj-Varjú & Salomaa (1997) and Csuhaj-Varjú & Mitrana (2000). The topic was introduced in Castellanos et al. (2003) and Martín-Vide et al. (2003), and further developed in Castellanos et al. (2005), Csuhaj-Varjú et al. (2005).

With this background and theoretical connections, it is easy to understand how NEPs can be described as agential bio-inspired context-sensitive systems. Many disciplines are needed of these types of models that are able to support a biological framework in a collaborative environment. The conjunction of these features allows applying the system to a number of areas, beyond generation and recognition in formal language theory. NLP is one of the fields with a lack of biological models and with a clear suitability for agential approaches.

NEPs have significant intrinsic multi-agent capabilities together with the environmental adaptability that is typical of bio-inspired models. Some of the characteristics of NEPs architecture are the following: *Modularization, contextualization* and *redefinition* of agent capabilities, s*ynchronization*, e*volvability* and *learnability*.

Inside of the construct, every agent is *autonomous, specialized, context-interactive* and *learning-capable*.

In what refers to the functioning of NEPs, two main features deserve to be highlighted: e*mergence* and p*arallelism.*

Because of those features, NEPs seems to be a suitable model for tackling natural languages. One of the main problems of natural language is that it is generated in the brain, and there is a lack of knowledge of the mental processes the mind undergoes to bring about a sentence. While expecting new advances in neuro-science, we have to use models that seem to fit better to NLP. Modularity has shown to be an important idea in a wide range of fields: cognitive science, computer science and, of course, NLP. NEPs provide a suitable theoretical framework for formalization of modularity in NLP.

Another chief problem for the formalization and processing of natural language is its changing nature. Not only words, but also rules, meaning and phonemes can take different shapes during the process of computation. Formal models based in mathematical language have a lack of flexibility to describe natural language. Biological models seem to be better to this task, since biological entities share with languages the concept of "evolution". From this perspective, NEPs offer enough flexibility to model any change at any moment in any part of the system. Besides, as a bio-inspired method of computation, they have the capability of simulating natural evolution in a highly pertinent and specialized way.

Some linguistic disciplines, as pragmatics or semantics, are context-driven areas, where the same utterance has different meanings in different contexts. To model such variation, a system with a good definition of environment is needed. NEPs offer some kind of solution to approach formal semantics and formal pragmatics from a natural computing perspective.

Finally, the multimodal approach to communication, where not just production, but also gestures, vision and supra-segmental features of sounds have to be tackled, refers to a parallel way of processing. NEPs allow modules to work in parallel. The autonomy of every one of the processors and the possible miscoordination between them can also give account of several problems of speech.

Examples of NEPs applications to NLP can be found in Bel-Enguix and Jiménez-López (2005c, 2006a).

FUTURE TRENDS

Three general formalisms for dealing with NLP by means of biological methods have been introduced,

focusing on the formal definition of several frameworks that adapt models coming from the area of bio-inspired computation to NLP needs. The main trends for the future focus on the implementation of these models in order to test their computational advantages over classical models of NLP without biological inspiration.

CONCLUSION

The coincidences between several structures of language and biology allow us, in the field of NLP, to take advantage of the bio-inspired models formalized by theoretical computer science. Moreover, the multi-agent capabilities of some of these models make them a suitable tool for simulating the processes of generation and recognition in natural language.

Biological methods coming from computer science can be very useful in the field of natural language, since they provide simple, flexible and intuitive tools for describing natural languages and making easier their implementation in NLP systems.

This research provides an integrative path for biology, computer science and NLP – three branches of human knowledge that have to be together in the development of new systems of communication for future global society.

REFERENCES

Adleman, L.M. (1994). Molecular Computation of Solutions to Combinatorial Problems. *Science*, 226, 1021-1024.

Asimov, I. (1968) . *Il Codice Genetico*. Torino: Einaudi.

Bel-Enguix, G. & Jiménez-López, M.D. (2005a). Byosyntax. An Overview. *Fundamenta Informaticae*, 64, 1-12.

Bel-Enguix, G. & Jiménez-López, M.D. (2005b). Linguistic Membrane Systems and Applications. In Gh. Ciobanu, Gh. Păun & M.J. Pérez-Jiménez (Eds.), *Applications of Membrane Computing* (pp. 347-388). Berlin: Springer.

Bel-Enguix, G. & Jiménez-López, M.D. (2005c). Analysing Sentences with Networks of Evolutionary Processors. In J. Mira & J.R. Álvarez (Eds.), *Artificial*

N

Intelligence and Knowledge Engineering Applications: A Bioinspired Approach (pp. 102-111). LNCS 3562. Berlin: Springer.

Bel-Enguix, G. & Jiménez-López, M.D. (2006a). Cognitive Modeling with Networks of Evolutionary Processors. A Preview. In J. Multisita & H. Haaparanta (Eds.), *Proceedings of the Workshop on Human Centered Technology HCT06* (pp. 268-275), Pori: Tampere University of Technology, Publication 6.

Bel-Enguix, G. & Jiménez-López, M.D. (2006b). Computing Dialogues with Membranes. *Electronic Notes on Theoretical Computer Science,* 157(4), 57-73.

Bel-Enguix, G. & Jiménez-López, M.D. (2007). Dynamic Meaning Membrane Systems: An Application to the Description of Semantic Change. *Fundamenta Informaticae,* 76(3), 219-237.

Bennett, C.H. (1973). Logical Reversibility of Computation. *IBM Journal of Research Development*, 17, 525-532.

Castellanos, J., Leupold, P. & Mitrana. V. (2005). Descriptional and Computational Complexity Aspects of Hybrid Networks of Evolutionary Processors. *Theoretical Computer Science*, 330(2), 205-220.

Castellanos, J., Martín-Vide, C., Mitrana, V. & Sempere, J.M. (2003). Networks of Evolutionary Processors. *Acta Informatica*, 39, 517-529.

Chomsky, N. (1956). Three Models for the Description of Language. *IRE Transactions on Information Theory*, 2, 113-124.

Collado Vides, J. (1989). A Transformation-Grammar Approach to the Study of Regulation of Gene Expression. *Journal of Theoretical Biology*, 136, 403–425.

Conrad, M. (1995). The Price of Programmability. In R. Herken (Ed.), *The Universal Turing Machine: A Half-Century Survey* (pp. 261-282). Wien: Springer.

Csuhaj-Varjú, E., Dassow, J., Kelemen, J. & Păun, Gh. (1994). *Grammar Systems.* London: Gordon and Breach.

Csuhaj-Varjú, E., Martín-Vide, C. & Mitrana, V. (2005). Hybrid Networks of Evolutionary Processors are Computational Complete. *Acta Informatica*, 41(4-5), 257-272.

Csuhaj-Varjú, E. & Mitrana, V. (2000). Evolutionary Systems: A Language Generating Device Inspired by Evolving Communities of Cells. *Acta Informatica*, 36, 913–926.

Csuhaj-Varjú, E. & Salomaa, A. (1997). Networks of Parallel Language Processors. In Gh. Păun & A. Salomaa (Eds.), *New Trends in Formal Languages* (pp. 299-318). LNCS 1218. Berlin: Springer.

Errico, L. & Jesshope, C. (1994). Towards a New Architecture for Symbolic Processing. In I. Plander (Ed.), *Artificial Intelligence and Information-Control Systems of Robots '94* (pp. 31-40). World Scientific Publisher.

Feynman, R.P. (1961). There's Plenty of Room at the Bottom. In D.H. Hilbert (Ed.), *Miniaturization* (pp. 282-296). Reinhold.

Forster, P. (1997). Network Analysis of Word Lists. In *Third International Conference on Quantitative Linguistics* (pp. 184-186). Helsinki: Research Institute for the Languages of Finland.

Hillis, W.D. (1985). *The Connection Machine*. Cambridge: MIT Press.

Jacob, F. (1970). *La Logique du Vivant. Une historie de l'heredité*. Paris: Gallimard.

Jakobson, R. (1970). Linguistics. In *Main Trends of Research in the Social and Human Sciences* (pp. 419-463). Paris: Mouton.

Jakobson, R. (1973). *Essais de Linguistique Générale. 2. Rapports Internes et Externes du Language*. Paris: Les Éditions de Minuit.

Ji, S. (2002). Microsemiotics of DNA. *Semiotica*, 138 (1/4), 15-42.

Marcus, S. (1998). Language at the Crossroad of Computation and Biology. In Gh. Păun (Ed.), *Computing with Bio-Molecules* (pp. 1-35). Singapore: Springer.

Martín-Vide, C., Mitrana, V., Pérez-Jiménez, M. & Sancho-Caparrini, F. (2003). Hybrid Networks of Evolutionary Processors. In E. Cantú-Paz et al. (Eds.), *Genetic and Evolutionary Computation* (Part I, pp. 401–412). LNCS 2723. Berlin: Springer.

Păun, Gh. (2000). Computing with Membranes. *Journal of Computer and System Sciences*, 61, 108-143

Păun, Gh., Rozenberg, G. & Salomaa, A. (1998). *DNA Computing. New Computing Paradigms*. Berlin: Springer.

Pawlak, Z. (1965). *Gramatyka i Matematika*. Warzsawa: Panstwowe Zakady Wydawnietw Szkolnych.

Sakakibara, Y., Brown, M., Underwood, R., Saira Mian, I. & Haussler, D. (1994). Stochastic Context-Free Grammars for Modeling RNA. In *Proceedings of the 27th Hawaii International Conference on System Sciences* (pp. 284-283). Honolulu: IEEE Computer Society Press.

Scheicher, A. (1863). *Die Darwinsche Theorie und die Sprachwissenschaft*. Weimar.

Schmidt, J. (1872), *Die Verwantschaftsverhältnisse der Indogermanischen Sprachen*. Weimar: Böhlau.

Searls, D. (1989). Investigating the Linguistics of DNA with Definite Clause Grammars. In E. Lusk & R. Overbeek (Eds.), *Logic Programming: Proceedings of the North American Conference on Logic Programming* (vol. 1, pp. 189-208). Association for Logic Programming.

Searls, D. (1993). The Linguistics of DNA. *American Scientist*, 80, 579–591.

Uemura, Y., Hasegawa, A., Kobayashi, S. & Yokomori, T. (1999). Tree Adjoining Grammars for RNA Structure Prediction. *Theoretical Computer Science*, 210(2), 277-303.

Watson, J.D. (1968). *Biologie Moléculaire du Gène*. Paris: Ediscience.

KEY TERMS

Grammar Systems Theory: A consolidated and active branch in the field of formal languages that provides syntactic models for describing multi-agent systems at the symbolic level using tools from formal languages and grammars.

Membrane Systems: In a membrane system multisets of objects are placed in the compartments defined by the membrane structure that delimits the system from its environment. Each membrane identifies a *region*, the space between it and all directly inner membranes. Objects evolve by means of reaction rules associated with compartments, and applied in a maximally parallel, nondeterministic manner. Objects can pass through membranes, membranes can change their permeability, dissolve and divide.

Multi-Agent System: A system composed of a set of computational agents that perform local problem solving and cooperatively interact to solve a single problem (or reach a goal) difficult to be solve (achieved) by an individual agent.

Mutations: Several types of transformations in a single string.

Natural Computing: Research field that deals with computational techniques inspired by nature and natural systems. This type of computing includes evolutionary algorithms, neural networks, molecular computing and quantum computing.

Neural Network: Interconnected group of artificial neurons that uses a mathematical or a computational model for information processing based on a connectionist approach to computation. It involves a network of simple processing elements that can exhibit complex global behaviour.

Splicing: Operation which consists of splitting up two strings in an arbitrary way and sticking the left side of the first one to the right side of the second one (direct splicing), and the left side of the second one to the right side of the first one (inverse splicing).

Natural Language Understanding and Assessment

Vasile Rus
The University of Memphis, USA

Philip M. McCarthy
The University of Memphis, USA

Danielle S. McNamara
The University of Memphis, USA

Arthur C. Graesser
The University of Memphis, USA

INTRODUCTION

Natural language understanding and assessment is a subset of natural language processing (NLP). The primary purpose of natural language understanding algorithms is to convert written or spoken human language into representations that can be manipulated by computer programs. Complex learning environments such as intelligent tutoring systems (ITSs) often depend on natural language understanding for fast and accurate interpretation of human language so that the system can respond intelligently in natural language. These ITSs function by interpreting the meaning of student input, assessing the extent to which it manifests learning, and generating suitable feedback to the learner. To operate effectively, systems need to be fast enough to operate in the real time environments of ITSs. Delays in feedback caused by computational processing run the risk of frustrating the user and leading to lower engagement with the system. At the same time, the accuracy of assessing student input is critical because inaccurate feedback can potentially compromise learning and lower the student's motivation and metacognitive awareness of the learning goals of the system (Millis et al., 2007). As such, student input in ITSs requires an assessment approach that is fast enough to operate in real time but accurate enough to provide appropriate evaluation.

One of the ways in which ITSs with natural language understanding verify student input is through *matching*. In some cases, the match is between the user input and a pre-selected *stored answer to a question*, *solution to a problem*, *misconception*, or other form of *benchmark response*. In other cases, the system evaluates the degree to which the student input varies from a complex representation or a dynamically computed structure. The computation of matches and similarity metrics are limited by the fidelity and flexibility of the computational linguistics modules.

The major challenge with assessing natural language input is that it is relatively unconstrained and rarely follows brittle rules in its computation of spelling, syntax, and semantics (McCarthy et al., 2007). Researchers who have developed tutorial dialogue systems in natural language have explored the accuracy of matching students' written input to targeted knowledge. Examples of these systems are AutoTutor and Why-Atlas, which tutor students on Newtonian physics (Graesser, Olney, Haynes, & Chipman, 2005; VanLehn , Graesser, et al., 2007), and the iSTART system, which helps students read text at deeper levels (McNamara, Levinstein, & Boonthum, 2004). Systems such as these have typically relied on statistical representations, such as latent semantic analysis (LSA; Landauer, McNamara, Dennis, & Kintsch, 2007) and content word overlap metrics (McNamara, Boonthum, et al., 2007). Indeed, such statistical and word overlap algorithms can boast much success. However, over short dialogue exchanges (such as those in ITSs), the accuracy of interpretation can be seriously compromised without a deeper level of lexico-syntactic textual assessment (McCarthy et al., 2007). Such a lexico-syntactic approach, *entailment evaluation*, is presented in this chapter. The approach incorporates deeper natural language processing solutions for ITSs with natural language exchanges while

remaining sufficiently fast to provide real time assessment of user input.

BACKGROUND

Entailment evaluations help in the assessment of the appropriateness of student responses during ITS exchanges. Entailment can be distinguished from three similar terms (implicature, paraphrase, and elaboration), all of which are also important for assessment in ITS environments (McCarthy et al, 2007).

The terms *entailment* is often associated with the highly similar concept of *implicature*. The distinction is that entailment is reserved for linguistic-based inferences that are closely tied to *explicit* words, syntactic constructions, and formal semantics, as opposed to the knowledge-based *implied* referents and references, for which the term implicature is more appropriate (McCarthy et al., 2007). Implicature corresponds to the controlled knowledge-based elaborative inferences defined by Kintsch (1993) or to knowledge-based inferences defined in the inference taxonomies in discourse psychology (Graesser, Singer, & Trabasso, 1994).

The terms *paraphrase* and *elaboration* also need to be distinguished from entailment. A *paraphrase* is a reasonable restatement of the text. Thus, a paraphrase is a form of entailment, yet an entailment is not necessarily a paraphrase. This asymmetric relation can be understood if we consider that *John went to the store* is entailed by (but not a paraphrase of) *John drove to the store to buy supplies*. The term *elaboration* refers to information that is generated inferentially or associatively in response to the text being analyzed, but without the systematic and sometimes formal constraints of entailment, implicature, or paraphrase. Examples of each term are provided below for the sentence *John drove to the store to buy supplies*.

Entailment: *John went to the store.*
(Explicit, logical implication based on the text)

Implicature: *John bought some supplies.*
(Implicit, reasonable assumption from the text, although not explicitly stated in the text)

Paraphrase: *He took his car to the store to get things that he wanted.*

(Reasonable re-statement of all and only the critical information in the text)

Elaboration: *He could have borrowed stuff.*
(Reasonable *reaction* to the text)

Evaluating entailment is generally referred to as the task of *recognizing textual entailment* (RTE; Dagan, Glickman, & Magnini, 2005). Specifically, it is the task of deciding, given two text fragments, whether the meaning of one text logically infers the other. When it does, the evaluation is deemed as T (the entailing text) entails H (the entailed hypothesis). For example, a text (from the RTE data) of *Eyeing the huge market potential, currently led by Google, Yahoo took over search company Overture Services Inc last year* would entail a hypothesis of *Yahoo bought Overture*. The task of recognizing entailment is relevant to a large number of applications, including machine translation, question answering, and information retrieval.

The task of textual entailment has been a priority in investigations of information retrieval (Monz & de Rijke, 2001) and automated language processing (Pazienza, Pennacchiotti, & Zanzotto, 2005). In related work, Moldovan and Rus (2001) analyzed how to use unification and matching to address the *answer correctness* problem. Similar to entailment, *answer correctness* is the task of deciding whether candidate answers logically imply an ideal answer to a question.

THE LEXICO-SYNTACTIC ENTAILMENT APPROACH

A complete solution to the textual entailment challenge requires linguistic information, reasoning, and world knowledge (Rus, McCarthy, McNamara, & Graesser, in press). This chapter focuses on the role of linguistic information in making entailment decisions. The overall goal is to produce a light (i.e. computationally inexpensive), but accurate solution that could be used in interactive systems such as ITSs. Solutions that rely on processing-intensive deep representations (e.g., frame semantics and reasoning) and large structured repositories of information (e.g., ResearchCyc) are impractical for interactive tasks because they result in lengthy response times, causing user dissatisfaction.

One solution for recognizing textual entailment is based on subsumption. In general, an object X subsumes

an object Y if and only if X is more general than or identical to Y. Applied to textual entailment, subsumption translates as follows: hypothesis H is entailed from T if and only if T subsumes H. The solution has two phases: (I) map both T and H into graph structures and (II) perform a subsumption operation between the T-graph and H-graph. An entailment score, entail(T,H), is computed, quantifying the degree to which the T-graph subsumes the H-graph.

In *phase I*, the two text fragments involved in a textual entailment decision are initially mapped onto a graph representation. The graph representation employed is based on the dependency-graph formalisms of Mel'cuk (1998). The mapping relies on information from syntactic parse trees. A phrase-based parser is used to derive the dependencies. Although a dependency-parser may be adopted, our particular research agenda required partial phrase parsers for other tasks such as computing cohesion metrics. Having a phrase-based and dependency parser integrated in the system would have led to a heavier, less interactive system. A parse tree groups words into phrases and organizes these phrases into hierarchical tree structures from which syntactic dependencies among concepts can be detected. The system uses Charniak's (2000) parser to obtain parse trees and Magerman's (1994) head-detection rules to obtain the head of each phrase. A dependency tree is generated by linking the head of each phrase to its modifiers in a systematic mapping process. The dependency tree encodes exclusively local dependencies (head-modifiers), as opposed to long-distance (remote) dependencies, such as the remote subject relation between *bombers* and *enter* in the sentence *The bombers managed to enter the embassy compounds*. Thus, in this stage, the dependency tree is transformed onto a *dependency graph* by generating *remote dependencies* between content words. Remote dependencies are computed by a naive-Bayes functional tagger (Rus

& Desai, 2005). An example of a dependency graph is shown in Figure 1 for the sentence *The two objects will cover the same horizontal distance*. For instance, there is a subject (*subj)* dependency relation between *objects* and *cover*.

In *phase II*, the textual entailment problem (i.e., each T and H) is mapped into a specific example of graph isomorphism called *subsumption* (also known as *containment*). Isomorphism in graph theory addresses the problem of testing whether two graphs are the same.

A graph G = (V, E) consists of a set of nodes or *vertices* V and a set of *edges* E. Graphs can be used to model the linguistic information embedded in a sentence: vertices represent concepts (e.g., *bombers*, *joint venture*) and edges represent syntactic relations among concepts (e.g., the edge labeled *subj* connects the verb *cover* to its subject *objects* in Figure 1). The Text (T) entails the Hypothesis (H) if and only if the hypothesis graph is subsumed (or contained) by the text graph.

The subsumption algorithm for textual entailment (Rus et al., in press) has three major steps: (1) find an isomorphism between V_H (set of vertices of the Hypothesis graph) and V_T; (2) check whether the labeled edges in H, E_H, have correspondents in E_T; and (3) compute score. In step 1, for each vertex V_H, a correspondent V_T node is sought. If a vertex in H does not have a direct correspondent in T, a thesaurus is used to find all possible synonyms for vertices. Step 2 takes each relation in H and checks its presence in T. The checking is augmented with relation equivalences among linguistic phenomena such as possessives and linking verbs (e.g. *be, have*). For instance, *tall man* would be equivalent to *man is tall*. A normalized score for vertices and edge mapping is then computed. The score for the entire entailment is the sum of each individual vertex and edge matching score. Finally, the score must account for negation. The approach handles both *explicit* and *implicit* negation. Explicit negation is indicated by particles such as *no, not, neither … nor*

Figure 1. An example of a dependency graph

The two objects will cover the same horizontal distance.
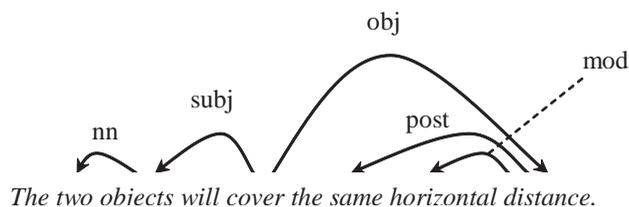
and the shortened form *n't*. Implicit negation is present in text via deeper lexico-semantic relations among linguistic expressions. The most obvious example is the *antonymy* relation among words, which is retrieved from WordNet (Miller, 1995). Negation is accommodated in the score after making the entailment decision for the Text-Hypothesis pair (without negation). If any one of the text fragments is negated, the decision is reversed, but if both are negated the decision is retained (double-negation), and so forth.

Entailment for Intelligent Tutoring Systems

The problem of evaluating student input in ITSs with natural language understanding is modeled here as a textual entailment problem. Results of this approach are shown on data sets from two ITSs: AutoTutor and iSTART. Data from the AutoTutor experiments involve college students learning Newtonian physics, whereas data from iSTART involve adolescent and college students constructing explanations about science texts.

AutoTutor

AutoTutor (autotutor.org) teaches topics such as Newtonian physics, computer literacy, and critical thinking by holding a dialogue in natural language with the student. The system presents deep-reasoning questions to the student that call for explanations or other elaborate answers. AutoTutor has a list of anticipated good answers (or *expectations*) and a list of *misconceptions* associated with each main question. AutoTutor guides the student in articulating the expectations through a number of dialogue moves and adaptively responds to the student by giving short *feedback* on the quality of student contributions.

To understand how the entailment approach helps to assess the appropriateness of student responses in AutoTutor, consider the following AutoTutor problem:

Suppose a runner is running in a straight line at constant speed, and the runner throws a pumpkin straight up. Where will the pumpkin land? Explain why.

An expectation for this problem is *The object will continue to move at the same horizontal velocity as the person when it is thrown*. A real student answer is *The pumpkin and the runner have the same horizontal velocity before and after release*. The expert judgment of this response was *very good*. Such *expectation/student-input* (E-S) pairs can be viewed as an entailment pair of Text-Hypothesis. The task is to find the truth value of the student answer based on the true fact encoded in the expectation. Rus and Graesser (2006) examined how the lexico-syntactic system described in the previous section performed on a test set of 125 E-S pairs collected from a sample of AutoTutor tutorial dialogues. The lexico-syntactic approach provided the best accuracy (69%), whereas a Latent Semantic Analysis (LSA, Landauer et al., 2007) approach yielded an accuracy of 60%. Such a result illustrates the value of augmenting AutoTutor with lexico-syntactic natural language understanding.

iSTART (Interactive Strategy Trainer for Active Reading and Thinking)

The primary goal of iSTART (istartreading.com) is to help high school and college students learn to use reading comprehension strategies that support deeper understanding. iSTART's design combines the power of self-explanation in facilitating deep learning (McNamara et al., 2004) with content-sensitive, interactive strategy training. The iSTART system helps students learn to self-explain using a variety of reading strategies (e.g., rewording the text, or *paraphrasing*; or *elaborating* on the text by linking textual content to what the reader already knows). The final stage of the iSTART process requires students to self-explain sentences from two short passages. Scaffolded feedback is provided to the students based on the quality of the student responses.

The entailment evaluation has been used in two iSTART studies. In Rus et al. (2007), a corpus of iSTART self-explanation responses was evaluated by an array of textual evaluation measures. The results demonstrated that the entailment approach was the most powerful distinguishing index of the self-explanation categories (Entailer: $F(1,1228) = 25.05, p < .001$; LSA: $F(1,1228) = 2.98, p > .01$). In McCarthy et al. (2007), iSTART self explanations were hand-coded for degree of entailment, paraphrase, versus elaboration. Once again, the entailment evaluation proved to be a more powerful predictor of these categories than traditional measures: for *entailment*, the Entailer was a significant predictor ($t= 9.61, p < .001$) and LSA was a marginal predictor ($t = -1.90, p = .061$); for *elaboration* and for

paraphrase the Entailer was again a significant predictor ($t = -7.98$, $p < .001$; $t = 5.62$, $p < .001$, respectively), whereas LSA results were not significant.

FUTURE TRENDS

While the results of the entailment evaluation have been encouraging, a variety of developments of the approach are underway. For example, there are plans to weight words by their specificity and to learn syntactic patterns or transformations that lead to similar meanings. The current negation detection algorithm will be extended to assess plausible implicit forms of negation in words such as *denied*, *denies*, *without*, *ruled out.* A second extension addresses issues of relative opposites: knowing that an object is *not* hot does not entail that the object is cold (i.e., it could simply be warm).

CONCLUSION

Recognizing and assessing textual entailment is a prominent and challenging task in the fields of Natural Language Processing and Artificial Intelligence. This chapter presented a lexico-syntactic approach to the task of evaluating entailment. The approach is light, using minimal knowledge resources, yet it has delivered high performance in evaluations of three data sets involving natural language interactions in ITSs. The entailment approach is a promising step in achieving the goal of fast and effective evaluation of student contributions in short text exchanges, which is needed to provide optimal feedback and responses to student learners.

ACKNOWLEDGMENT

REFERENCES

Charniak, E., (2000). A maximum-entropy-inspired parser. In *Proceedings ANLP-NAACL'2000*, Seattle, Washington, 132-139.

Dagan, I., Glickman, O., & Magnini, B. (2005), The PASCAL recognising textual entailment challenge. In *Proceedings of the Recognizing Textual Entailment Challenge Workshop,* Southampton, U.K, 1-8.

Graesser, A. C., Olney, A., Haynes, B.C., & Chipman, P. (2005). AutoTutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue. In C. Forsythe, M. L. Bernard, and T. E. Goldsmith, (Eds.). *Cognitive Systems: Human Cognitive Models in Systems Design*. Erlbaum, Mahwah, NJ.

Graesser, A.C., & Singer, M. & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review*, 101, 371-95.

Kintsch, W. (1993). Information accretion and reduction in text processing: Inferences. *Discourse Processes*, 16:193-202, 1993.

Landauer, T., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.) (2007). *Handbook of Latent Semantic Analysis*. Mahwah, NJ: Erlbaum.

Magerman, D.M., (1994). Natural language parsing as statistical pattern recognition. Ph.D. thesis, Stanford University, February.

McCarthy, P.M., Rus, V., Crossley, S.A., Bigham, S.C., Graesser, A.C., & McNamara, D.S. (2007). Assessing entailer with a corpus of natural language. In *Proceedings of the Florida Artificial Intelligence Research Society International Conference (FLAIRS)* (pp. 247-252), Menlo Park, CA: AAAI Press.

McNamara, D. S., Boonthum, C., Levinstein, I. B., & Millis, K. (2007). Evaluating self-explanations in iSTART: comparing word-based and LSA algorithms. In Landauer, T., D.S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of LSA*. Mahwah, NJ: Erlbaum, 227-241.

McNamara, D. S., Levinstein, I. B. & Boonthum, C. (2004). iSTART: Interactive strategy trainer for active reading and thinking. *Behavioral Research Methods, Instruments, & Computers*, *36*, 222-233.

N

Miller, G. (1995), Wordnet: a lexical database for English. *Communications of the ACM*, 38, 39-41.

Millis, K., Magliano, J., Wiemer-Hastings, K., Todaro, S., & McNamara, D.S. (2007). Assessing and improving comprehension with Latent Semantic Analysis. In Landauer, T., D.S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of Latent Semantic Analysis* (pp. 207-225). Mahwah, NJ: Erlbaum.

Mel'cuk, I.A. (1998). Dependency syntax: Theory and practice. State University of New York Press, Albany, NY.

Moldovan, D.I. & Rus, V. (2001) Logic form transformation of WordNet and its applicability to question answering. *Proceedings of the ACL 2001 Conference*, Toulouse, France, 394-401.

Monz, C. & de Rijke, M. (2001). *Light-weight entailment checking for computational semantics*. In P. Blackburn and M. Kohlhase (Eds.), *Proceedings of Inference in Computational Semantics* (ICoS-3), Siena, Italy (pp. 59-72).

Pazienza, M.T., Pennacchiotti, M., & Zanzotto, F. M. (2005). Textual entailment as syntactic graph distance: A rule based and SVM based approach. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*, Southampton, U.K., April 11 – 13, 25-28.

Rus, V. & Desai, K. (2005). Assigning function tags with a simple model. In *Proceedings of Conference on Intelligent Text Processing and Computational Linguistics (CICLing),* Mexico City, Mexico, 112-115.

Rus, V., McCarthy, P.M., McNamara, D.S., & Graesser, A.C. (in press). A study of textual entailment, *International Journal on Artificial Intelligence Tools*.

Rus, V., Graesser, A. C., & Desai, K. (2005). Lexico-Syntactic subsumption for textual entailment. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, Borovets, Bulgaria, 444-452.

Rus, V., McCarthy, P.M., McNamara, D.S., & Graesser, A.C. (2007). *Assessing student self-explanations in an intelligent tutoring system*. In D. S. McNamara & J. G. Trafton (Eds.), *Proceedings of the 29th Annual Cognitive Science Society*. Austin, TX: Cognitive Science Society.

VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., & Rose, C.P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, 31, 3-62.

KEY TERMS

Dependency: Binary relations between words in a sentence whose label indicates the syntactic relation among the two words.

Entailment: The task of deciding whether a text fragment logically or semantically infers another text fragment.

Expectation: A stored (generally *ideal*) answer to a problem, against which input is evaluated; concept used in ITSs.

Graph Subsumption: A specific example of graph isomorphism. Isomorphism exists when two graphs are equivalent. Subsumption can be viewed as subgraph isomorphism.

Intelligent Tutoring System: Interactive, feedback-based computer systems designed to help students learn various topics.

Latent Semantic Analysis: A statistical technique for human language understanding based on words that co-occur in documents of large corpora.

Natural Language Processing: The science of capturing the meaning of human language in computational representations and algorithms.

Natural Language Understanding and Assessment: An NLP subset focusing on evaluating natural language input in intelligent tutoring systems.

Syntactic Parsing: The process of discovering the underlying structure of sentences.

Navigation by Image–Based Visual Homing

Matthew Szenher
University of Edinburgh, UK

INTRODUCTION

Almost all autonomous robots need to navigate. We define navigation as do Franz & Mallot (2000): "Navigation is the process of determining and maintaining a course or trajectory to a goal location" (p. 134). We allow that this definition may be more restrictive than some readers are used to - it does not for example include problems like obstacle avoidance and position tracking - but it suits our purposes here.

Most algorithms published in the robotics literature localise in order to navigate (see e.g. Leonard & Durrant-Whyte (1991a)). That is, they determine their own location and the position of the goal in some suitable coordinate system. This approach is problematic for several reasons. Localisation requires a map of available landmarks (i.e. a list of landmark locations in some suitable coordinate system) and a description of those landmarks. In early work, the human operator provided the robot with a map of its environment. Researchers have recently, though, developed simultaneous localisation and mapping (SLAM) algorithms which allow robots to learn environmental maps while navigating (Leonard & Durrant-Whyte (1991b)). Of course, autonomous SLAM algorithms must choose which landmarks to map and sense these landmarks from a variety of different positions and orientations. Given a map, the robot has to associate sensed landmarks with those on the map. This data association problem is difficult in cluttered real-world environments and is an area of active research.

We describe in this chapter an alternative approach to navigation called visual homing which makes no explicit attempt to localise and thus requires no landmark map. There are broadly two types of visual homing algorithms: feature-based and image-based. The feature-based algorithms, as the name implies, attempt to extract the same features from multiple images and use the change in the appearance of corresponding features to navigate. Feature correspondence is - like data association - a difficult, open problem in real-world environments. We argue that image-based homing algorithms, which provide navigation information based on whole-image comparisons, are more suitable for real-world environments in contemporary robotics.

BACKGROUND

Visual homing algorithms make no attempt to localise in order to navigate. No map is therefore required. Instead, an image I_S (usually called a snapshot for historical reasons) is captured at a goal location $S = (x_S, y_S)$. Note that though S is defined as a point on a plane, most homing algorithms can be easily extended to three dimensions (see e.g. Zeil et al. (2003)). When a homing robot seeks to return to S from a nearby position $C = (x_C, y_C)$, it takes an image I_C and compares it with I_S. The home vector $H = S - C$ is inferred from the disparity between I_S and I_C (vectors are in upper case and bold in this work). The robot's orientation at C and S is often different; if this is the case, image disparity is meaningful only if I_C is rotated to account for this difference. Visual homing algorithms differ in how this disparity is computed.

Visual homing is an iterative process. The home vector H is frequently inaccurate, leading the robot closer to the goal position but not directly to it. If H does not take the robot to the goal, another image I_C is taken at the robot's new position and the process is repeated.

The images I_S and I_C are typically panoramic grayscale images. Panoramic images are useful because, for a given location (x,y) they contain the same image information regardless of the robot's orientation. Most researchers use a camera imaging a hemispheric, conical or paraboloid mirror to create these images (see e.g. Nayar (1997)).

Some visual homing algorithms extract features from I_S and I_C and use these to compute image disparity. Alternatively, disparity can be computed from entire images, essentially treating each pixel as a viable feature. Both feature-based and image-based visual homing algorithms are discussed below.

FEATURE-BASED VISUAL HOMING

Feature-based visual homing methods segment I_S and I_C into features and background (the feature extraction problem). Each identified feature in the snapshot is then usually paired with one feature in I_C (the correspondence problem). The home vector is inferred from - depending on the algorithm - the change in the bearing and/or apparent size of the paired features. Generally, in order for feature-based homing algorithms to work properly, they must reliably solve the feature extraction and correspondence problems.

The Snapshot Model (Cartwright & Collett (1983)) - the first visual homing algorithm to appear in the literature and the source of the term "snapshot" to describe the goal image - matches each snapshot feature with the current feature closest in bearing (after both images are rotated to the same external compass orientation). Features in (Cartwright & Collett (1983)) were black cylinders in an otherwise empty environment. Two unit vectors, one radial and the other tangential, are associated with each feature pair. The radial vector is parallel to the bearing of the snapshot feature; the tangential vector is perpendicular to the radial vector. The direction of the radial vector is chosen to move the agent so as to reduce the discrepancy in apparent size between paired features. The direction of the tangential vector is chosen to move the agent so as to reduce the discrepancy in bearing between paired features. The radial and tangential vectors for all feature pairs are averaged to produce a homing vector. The Snapshot Model was devised to explain the behaviour of nest-seeking honeybees but has inspired several robotic visual homing algorithms.

One such algorithm is the Average Landmark Vector (ALV) Model (Möller et al. (2001)). The ALV Model, like the Snapshot Model, extracts features from both I_C and I_S. The ALV Model, though, does not explicitly solve the correspondence problem. Instead, given features extracted from I_S, the algorithm computes and stores a unit vector ALV_S in the direction of the mean bearing to all features as seen from S. At C, the algorithm extracts features from I_C and computes their mean bearing, encoded in the unit vector ALV_C. The home vector **H** is defined as $ALV_C - ALV_S$. Figure 1 illustrates home vector computation for a simple environment with four easily discernible landmarks.

Several other interesting feature-based homing algorithms can be found in the literature. Unfortunately, space constraints prevent us from reviewing them here.

Two algorithms of note are: visual homing by "surfing the epipoles" (Basri et al. (1998)) and the Proportional Vector Model (Lambrinos et al. (2000)).

The Snapshot and ALV Models were tested by their creators in environments in which features contrasted highly with background and so were easy to extract. How is feature extraction and correspondence solved in real-world cluttered environments? One method is described in Gourichon et al. (2002). The authors use images converted to the HSV (Hue-Saturation-Value) colour space which is reported to be more resilient to illumination change than RGB. Features are defined as image regions of approximately equal colour (identified using a computationally expensive region-growing technique). Potential feature pairs are scored on their difference in average hue, average saturation, average intensity and bearing. The algorithm searches for a set of pairings which maximise the sum of individual match scores. The pairing scheme requires $O(n^2)$ pair-score computations (where n is the number of features). The algorithm is sometimes fooled by features with similar colours (specifically, pairing a blue chair in the snapshot image with a blue door in the current image). Gourichon et al. did not explore environments with changing lighting conditions.

Several other methods feature extraction and correspondence algorithms appear in the literature; see e.g. Rizzi et al. (2001), Lehrer & Bianco (2000) and Gaussier et al. (2000). Many of these suffer from some of the same problems as the algorithm of Gourichon et al. described above. The appearance of several competing feature extraction and correspondence algorithms in recent publications indicates that these are open and difficult problems; this is why we are advocating image-based homing in this chapter.

Figure 1. Illustration of Average Landmark Vector computation. See Section titled "Feature-based Visual Homing" for details
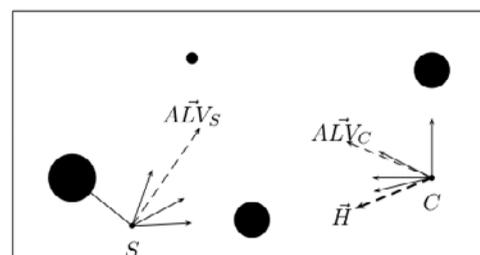
IMAGE-BASED VISUAL HOMING

Feature-based visual homing algorithms require consistent feature extraction and correspondence over a variety of viewing positions. Both of these are still open problems in computer vision. Existing solutions are often computationally intensive. Image-based visual homing algorithms avoid these problems altogether. They infer image disparity from entire images; no pixel is disregarded. We believe that these algorithms present a more viable option for real-world, real-time robotics.

Three image-based visual homing algorithms have been published so far; we describe these below.

Image Warping

The image warping algorithm (Franz et al. (1998)) asks the following question: When the robot is at C in some unknown orientation, what change in orientation and position is required to transform I_C into I_S? The robot needs to know the distance to all imaged objects in I_S to answer this question precisely. Not having this information, the image warping algorithm makes the assumption that all objects are at an equal (though unknown) distance from S. The algorithm searches for the values of position and orientation change which minimises the mean-square error between a transformed I_C and I_S. Since the mean square error function is rife with local minima, the authors resort to a brute force search over all permissible values of position and orientation change.

Unlikely as the equal distance assumption is, the algorithm frequently results in quite accurate values for H. Unlike most visual homing schemes, image warping requires no external compass reference. Unfortunately, the brute force search for the homing vector and the large number of transformations of I_C carried out during this search make image warping quite computationally expensive.

Homing with Optic Flow Techniques

When an imaging system moves from S to C, the image of a particular point in space moves from $I_S(x,y)$ to $I_C(x',y')$. This movement is called optic flow and $(x-x', y-y')$ is the so called pixel displacement vector. Vardy & Möller (2005) demonstrate that the home vector H can be inferred from a single displacement vector so long as the navigating robot is constrained to move on

a single plane. Several noisy displacement vectors can be combined to estimate H.

Vardy & Möller (2005) describe a number of methods, adapted from the optic flow literature, to estimate the displacement vector. One of the most successful methods – BlockMatch - segments the snapshot image into several equal-sized subimages. The algorithm then does a brute force search of a subset of I_C to find the best match for each subimage. A displacement vector is computed from the centre of each subimage to the centre of its match pair in I_C.

A less computationally intensive algorithm estimates the displacement vector from the intensity gradient at each pixel in I_C. The intensity gradient at a particular pixel can be computed straightforwardly from intensities surrounding that pixel. No brute-force search is required.

In comparative tests, Vardy & Möller demonstrated that their optic flow based methods perform consistently better than image warping in several unadulterated indoor environments. A drawback to the optic flow homing methods is that the robot is constrained to move on a single plane. The authors do not provide a way to extend their algorithm to three dimensional visual homing.

Surfing the Difference Surface

Zeil et al. (2003) describe a property of natural scenes which can be exploited for visual homing: as the Euclidean distance between S and C increases, the pixel-by-pixel root mean square (RMS) difference between I_S and I_C increases smoothly and monotonically. Labrosse and Mitchell discovered this phenomenon as well; see Mitchell & Labrosse (2004). Zeil et al. reported that the increase in the RMS signal was discernible from noise up to about three meters from S in their outdoor test environment; they call this region the catchment area.

RMS, when evaluated at locations in a subset of the plane surrounding S, forms a mathematical surface, the difference surface. A sample difference surface is shown in Figure 2(a) (see caption for details).

Zeil et al. describe a simple algorithm to home using the RMS difference surface. Their "Run-Down" algorithm directs the robot to move in its current direction while periodically sampling the RMS signal. When the current sample is greater than the previous, the robot is made to stop and turn ninety degrees (clockwise or

Figure 2. Two difference surfaces formed using the RMS image similarity measure. Both the surfaces and their contours are shown. In each case, the snapshot I_S was captured at x=150cm, y=150cm in a laboratory environment. (a) The snapshot was captured in the same illumination conditions as all other images. Notice the global minimum at the goal location and the absence of local minima. (b) Here we use the same snapshot image as in (a) but the lighting source has changed in all other images. The global minimum no longer appears at the goal location. When different goal locations were used, we observed qualitatively similar disturbances in the difference surfaces formed. The images used were taken from a database provided by Andrew Vardy which is described in Vardy & Möller (2005).

(a)

(b)
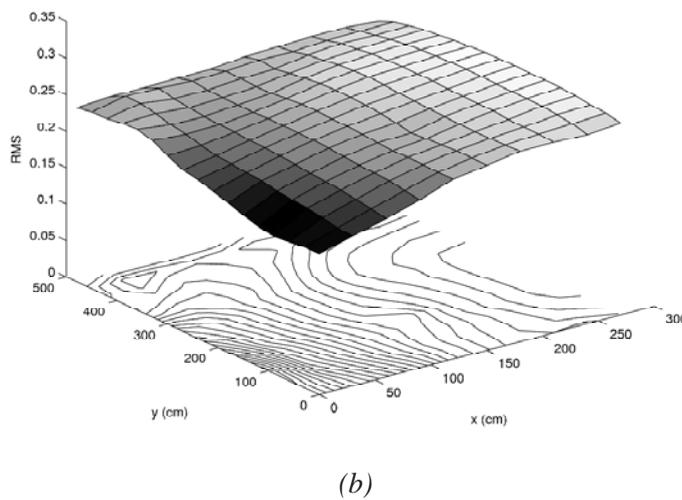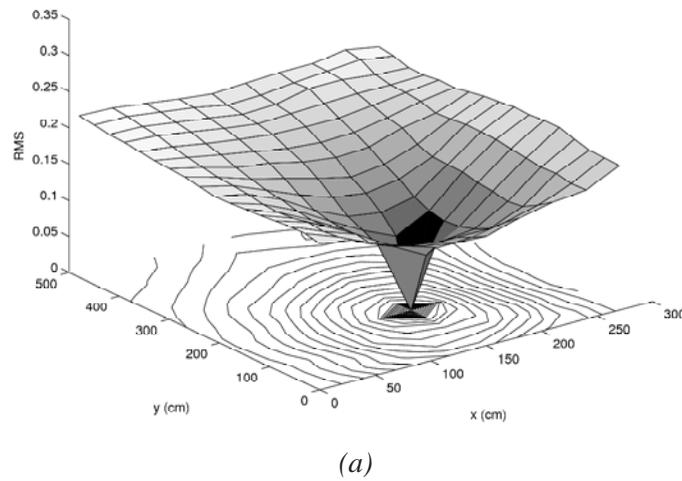
counter-clockwise, it does not matter). It then repeats the process in this new direction. The agent stops when the RMS signal falls below a pre-determined threshold. We have explored a biologically inspired difference surface homing method which was more successful than "Run-Down" in certain situations (Zampoglou et al. (2006)).

Unlike the optic flow methods described in the previous section, visual homing by optimising the difference surface is easily extensible to three dimensions (Zeil et al. (2003)).

Unfortunately, when lighting conditions change between capture of I_S and I_C , the minimum of the RMS difference surface often fails to coincide with S, making homing impossible (Figure 2(b)).

FUTURE TRENDS

No work has yet been published comparing the efficacy of the image-based homing algorithms described above. This would seem the logical next step for image-based homing researchers. As we mentioned in the section titled "Surfing the Difference Surface," the difference surface is disrupted by changes in lighting between captures of I_S and I_C. This problem obviously demands a solution and is a focus of our current research. Finally, it would be interesting to compare standard map-based navigation algorithms with the image-based visual homing methods presented here.

CONCLUSION

Visual homing algorithms - unlike most of the navigation algorithms found in the robotics literature - do not require a detailed map of their environment. This is because they make no attempt to explicitly infer their location with respect to the goal. These algorithms instead infer the home vector from the discrepancy between a stored snapshot image taken at the goal position and an image captured at their current location.

We reviewed two types of visual homing algorithms: feature-based and image-based. We argued that image-based algorithms are preferable because they make no attempt to solve the tough problems of consistent feature extraction and correspondence - solutions to which feature-based algorithms demand. Of the three

image-based algorithms reviewed, image warping is probably not practicable due to the computationally demanding brute force search required. Work is required to determine which of the two remaining image-based algorithms is more effective for robot homing in real-world environments.

REFERENCES

Basri, R., Rivlin, E., & Shimshoni, I. (1998). Visual homing: Surfing on the epipoles. In *The Proceedings of the Sixth International Conference on Computer Vision* (p. 863-869).

Cartwright, B., & Collett, T. (1983). Landmark learning in bees. *Journal of Comparative Physiology*, 151, 521-543.

Franz, M., & Mallot, H. (2000). Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30, 133-153.

Franz, M., Schölkopf, B., Mallot, H., & Bülthoff, H. (1998). Where did i take that snapshot? scene-based homing by image matching. *Biological Cybernetics*, 79, 191-202.

Gaussier, P., Joulain, C., Banquet, J., Leprêtre, S., & Revel, A. (2000). The visual homing problem: an example of robotics/biology cross fertilization. R*obotics and Autonomous Systems*, 30, 155-180.

Gourichon, S., Meyer, J., & Pirim, P. (2002). Using colored snapshots for short-range guidance in mobile robots. *International Journal of Robotics and Automation: Special Issue on Biologically Inspired Robotics*, 17 (4), 154-162.

Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., & Wehner, R. (2000). A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30, 39-64.

Lehrer, M., & Bianco, G. (2000). The turn-back-and-look behaviour: bee versus robot. *Biological Cybernetics*, 83, 211-229.

Leonard, J. J., & Durrant-Whyte, H. F. (1991a). Mobile robot localization by tracking geometric b e a c o n s . *IEEE Transactions on Robotics and Automation*, 7 (3), 376-382.

Leonard, J. J., & Durrant-Whyte, H. F. (1991b). Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems* (pp. 1442-1447). Osaka, Japan.

Möller, R., Lambrinos, D., Roggendorf, T., Pfeifer, R., & Wehner, R. (2001). Insect strategies of visual homing in mobile robots. In B. Webb & T. R. Consi (Eds.), *Biorobotics: Methods and Applications* (pp. 37-66). The MIT Press, Cambridge, Massachusetts.

Mitchell, T., & Labrosse, F. (2004). Visual homing: a purely appearance-based approach. In *Proceedings of TAROS (Towards Autonomous Robotic Systems)*. The University of Essex, UK.

Nayar, S. K. (1997). Omnidirectional video camera. In *Proceedings of DARPA image understanding workshop*. New Orleans, USA.

Rizzi, A., Duina, D., & Cassinis, R. (2001). A novel visual landmark matching for a biologically inspired homing. *Pattern Recognition Letters*, 22, 1371-1378.

Vardy, A., & Möller, R. (2005). Biologically plausible visual homing methods based on optical flow techniques. *Connection Science, Special Issue: Navigation*, 17 (1-2), 47-89.

Zampoglou, M., Szenher, M., & Webb, B. (2006). Adaptation of controllers for image-based homing. *Adaptive Behavior*, 14 (4), 381-399.

Zeil, J., Hofmann, M., & Chahl, J. (2003). Catchment areas of panoramic snapshots in outdoor scenes. *Journal of the Optical Society of America A*, 20 (3), 450-469.

KEY TERMS

Catchment Area: The area from which a goal location is reachable using a particular navigation algorithm.

Correspondence Problem: The problem of pairing an imaged feature extracted from one image with the same imaged feature extracted from a second image. The images may have been taken from different locations, changing the appearance of the features.

Image-based Visual Homing: Visual homing (see definition below) in which the home vector is estimated from the whole-image disparity between snapshot and current images. No feature extraction or correspondence is required.

Feature Extraction Problem: The problem of extracting the same imaged features from two images taken from (potentially) different locations.

Navigation: The process of determining and maintaining a course or trajectory to a goal location.

Optic Flow: The perceived movement of objects due to viewer translation and/or rotation.

Snapshot Image: In the visual homing literature, this is the image captured at the goal location.

Visual Homing: A method of navigating in which the relative location of the goal is inferred by comparing an image taken at the goal with the current image. No landmark map is required.

Nelder–Mead Evolutionary Hybrid Algorithms

Sanjoy Das
Kansas State University, USA

INTRODUCTION

Real world optimization problems are often too complex to be solved through analytic means. Evolutionary algorithms are a class of algorithms that borrow paradigms from nature to address them. These are stochastic methods of optimization that maintain a population of individual solutions, which correspond to points in the search space of the problem. These algorithms have been immensely popular as they are derivative-free techniques, are not as prone to getting trapped in local minima, and can be tailored specifically to suit any given problem. The performance of evolutionary algorithms can be improved further by adding a local search component to them. The Nelder-Mead simplex algorithm (Nelder & Mead, 1965) is a simple local search algorithm that has been routinely applied to improve the search process in evolutionary algorithms, and such a strategy has met with great success.

In this article, we provide an overview of the various strategies that have been adopted to hybridize two well-known evolutionary algorithms - genetic algorithms (GA) and particle swarm optimization (PSO).

BACKGROUND

Arguably, GAs are one of the most of all common population based approaches for optimization. The population of candidate solutions that these algorithms maintain in each generation are called chromosomes. GAs carry out the Darwinian operators of selection, mutation, and recombination, on these chromosomes, to perform their search (Mitchell, 1998). Each generation is improved by removing the poorer solutions from the population, while retaining the better ones, based on a fitness measure. This process is called selection. Following selection, a method of recombining solutions called crossover is applied. Here two (or more) parent solutions from the current generation are picked randomly for producing offspring to populate the next generation of solutions. The offspring chromosomes are then probabilistically subject to mutation, which is carried out by the addition of small random perturbations.

PSO is a more recent approach for optimization (Kennedy & Eberhart, 2001). Being modelled after the social behavior of organisms such as a flock of birds in flight or a school of fish swimming, it is considered an evolutionary algorithm only in a loose sense. Each solution within the population is called a particle in PSO. Each such particle's position in the search space is constantly updated within each generation, by the addition of the particle's velocity to it. The velocity of a particle is then adjusted towards the best position encountered in the particle's own history (individual best), as well as the best position in the current iteration (global best).

Since evolutionary algorithms use a population of individuals and randomized variational operators, they are adept at performing exploratory searches over their search spaces. However, when the aim is to produce outputs within reasonable time limits, it is important to balance this exploration with better exploitation of smaller-scale features in the fitness landscape. In the latter context, local search algorithms enable single solutions to be improved using local information (e.g., directional trends in fitness around each solution) and take the solution towards the closest maximum fitness. Hybrid algorithms that combine the advantages of exploration and exploitation comprise of a distinct area of evolutionary computation research that have been variously called as Lamarckian or memetic approaches, of which Nelder-Mead hybrids are a significant chunk.

NELDER-MEAD SIMPLEX BASED HYBRIDIZATION

The Nelder-Mead Downhill Simplex Algorithm

The Nelder-Mead simplex algorithm is a derivative-free local search technique that is capable of moving a cluster

Figure 1. Various operations in the Nelder-Mead simplex routine
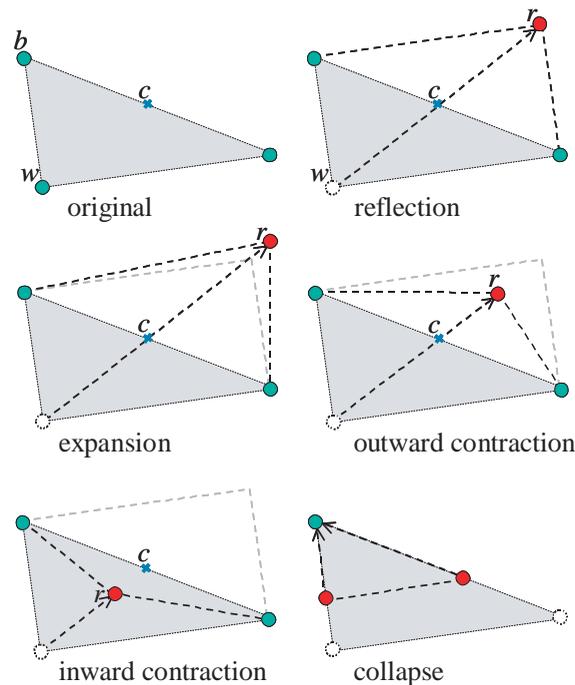
of solutions in the gradient direction and which, as per current research, can be very effectively combined with GA and PSO approaches. These hybrid evolutionary algorithms have been shown to be very successful in continuous optimization problems.

The Nelder-Mead smplex method makes use of a construct called a simplex (see Figure 1.). When the search space is *n*-dimensional, the simplex consists of *n*+1 solutions, s_i, $i = \{1, 2, \ldots, n+1\}$, that are usually closely spaced. As shown in the top left of Figure 1., in a two-dimensional search plane, a simplex is a triangle. The fitness of each solution is considered in each step of the Nelder-Mead method, and the worst solution *w* is identified. The centroid, *c*, of the remaining *n* points

$$c = \frac{1}{n}\sum s_i,$$

is computed and the reflection of *w* along it determined. This reflection yields a new solution *r* that replaces *w*, in the next step, as shown in the top right of Figure 1. If the solution *r* produced by this reflection has a higher fitness than any other solution in the simplex, the simplex is further expanded along the direction of

r, as shown in the middle left of the figure. On the other hand, if *r* has a low fitness compared to the others, the simplex is contracted. Contraction can be either outward or inward depending upon whether *r* is better or worse than *w*. The contraction operations are shown in the middle right and bottom left of the figure. If neither contraction improves the worst solution in the simplex, the best point in the simplex is computed, and a collapse is then carried out, and all the points of the simplex are moved a little closer towards the best one, as shown in the bottom right of the same figure.

The approaches taken to incorporate a simplex-based local search routine within the broad framework of a genetic algorithm fall under four different schemes that are shown in Figure 2. These are as follows:

Two-Phase Hybridization

This is the simplest of all approaches and has been applied to GAs (Chelouah & Siarry, 2000, Chelouah & Siarry, 2003, Robin, Orzati, Moreno, Homan & Bachtold, 2003). In the first phase in this scheme, a GA is applied to the optimization problem to explore the entire search space until one or more good solutions

are found, which can no longer be improved through the random operations of crossover and mutation. The Nelder-Mead simplex algorithm is then invoked in the second phase to further improve the solutions by allowing them to ascend towards their local maxima. In another approach the initial points of the simplex are obtained by taking the solution with the best fitness given by the GA, and then generating the remaining n points around it (Chelouah & Siarry, 2000, Chelouah & Siarry, 2003).

Serial Hybridization

In this scheme, the solutions in each generation are subject to the usual operators of the main evolutionary algorithm as well as the one or more steps of the Nelder-Mead simplex method. It has been successfully applied to hybridize GAs (Renders & Flasse, 1998, Yang & Douglas, 1998, Durand & Alliot, 1999, Guo & Shouyi, 2003, Trabia, 2004). This method has also been used in conjunction with PSO by Das *et al*. (Das, Koduru, Welch, Gui, Cochran, Wareing & Babin, 2006, Koduru, Welch, Das, 2007). In each generation, following the position and velocity updates, the population is

clustered into distinct clusters of $n+1$ solutions each, and a few steps of the Nelder-Mead algorithm applied separately to each cluster. The Nelder-Mead step is applied a fixed number of times per generation.

The serial hybridization scheme has been successfully implemented within a multi-objective optimization framework also (Koduru, Das, Welch 2007). Instead of fitness, a metric called fuzzy dominance is applied to discriminate between the $n+1$ solutions within a simplex. A solution that is not dominated by any other is assigned a fuzzy dominance of zero. The poorer a solution is, the higher the fuzzy dominance value it is assigned.

Parallel Hybridization

Such hybridization approaches assemble the offspring generation from the parent generation in two parallel tracks. The standard evolutionary algorithm operators are used to generate some of the offspring, while others are generated using the simplex algorithm. This strategy is applied to hybridize GAs (Yen, Liao, Lee & Randolph, 1998, Koduru, Das, Welch & Roe, 2004, Koduru, Das, Welch, Roe & Lopez-Dee, 2005). In these

Figure 2. Four different GA-simplex hybridization strategies
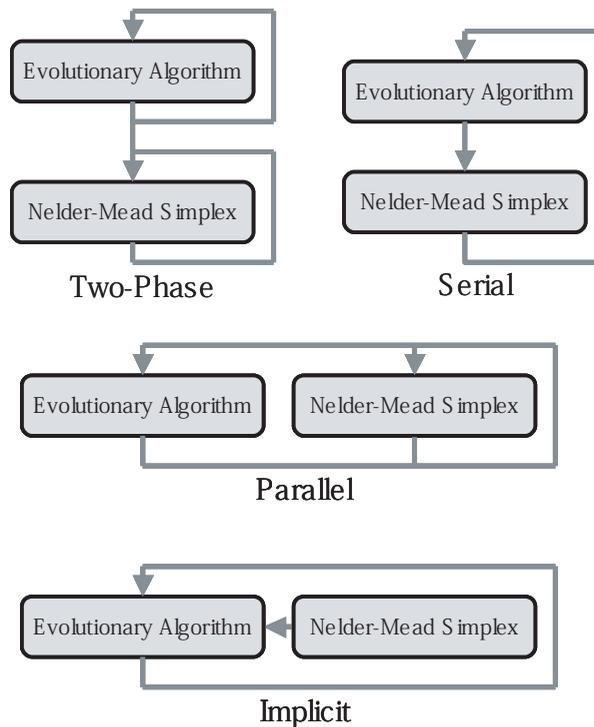
approaches, the best $n+1$ solutions (called elites) of each generation are picked to be improved further, using the Nelder-Mead simplex method. In another strategy, a probabilistic variant of the Nelder-Mead approach is used, where the amount of contraction and/or expansion of the simplex is determined randomly, but within specific limits (Yen, Liao, Lee & Randolph, 1998). The approaches taken in (Koduru, Das, Welch & Roe, 2004) and (Koduru, Das, Welch, Roe & Lopez-Dee, 2005) are multi-objective implementations that make use of the fuzzy dominance metric discussed earlier to identify the best and worst solutions. In order to preserve solution diversity within the population, the collapse operation is never used and the Nelder-Mead routine is terminated instead, when the need for one arises, within each generation.

This scheme has been used with PSO (Fan, Liang & Zahara 2004, Zahara, Fan & Tsai, 2005). As earlier, only the best $n+1$ points of the population are picked to undergo improvement using the Nelder-Mead simplex method. The remaining solutions in each generation are obtained using standard PSO position and velocity updates.

Implicit Hybridization

Here, the Nelder-Mead simplex algorithm is not applied directly. Instead, the approach is buried implicitly within any of the evolutionary algorithm's generic operators. One simple technique in GAs is the multi-parent simplex-based crossover (Renders & Bersini, 1994). This method applies a single Nelder-Mead step to produce a new offspring. Novel crossover techniques are also suggested (Bersini, 2002). In another method, each simplex is encoded as a chromosome and the algorithm uses a specially devised multi-parent crossover within the GA (Hedar & Fukushima, 2003).

Das *et al*. (Das, Koduru, Welch, Gui, Cochran, Wareing & Babin, 2006) use implicit hybridization in PSO by adding a term to the velocity of each particle that allows the latter to reorient its trajectory towards gradient direction sensed by the Nelder-Mead simplex (from the worst towards the centroid).

FUTURE TRENDS

Although traditionally, evolutionary algorithms have focussed on optimizing single objective functions, most practical problems in engineering are inherently multi-objective in nature. Consequently, multi-objective evolutionary optimization is a relatively new, emerging direction of evolutionary computation research. Perhaps the only attempts at incorporating Nelder-Mead simplex as an additional operator within a GA have been reported by Koduru, Das & Welch (*cf*. Koduru, Das, Welch, Roe, 2004). Clearly more research is required in this direction, and as multi-objective algorithms become more common, Nelder-Mead strategies will be investigated more vigorously.

PSO is a new technique for evolutionary optimization. Research into PSO-based hybrid algorithms has only recently begun to make its appearance. A few limited approaches have been suggested to hybridize PSO with Nelder-Mead simplex by Das *et al*. (Das, Koduru, Welch, Gui, Cochran, Wareing & Babin, 2006) and Zahara *et al*., (Zahara, Fan & Tsai, 2005). The method suggested in (Koduru, Das, Welch 2007) is, to the best of the author's knowledge, the only attempt at producing a multi-objective PSO hybrid algorithm. Here again, further investigation is necessary.

Although research into these evolutionary hybrid algorithms is over a decade old, with several good approaches having been suggested, there is no clear consensus about which approach is best suited for any given application. More research in this direction is warranted to obtain further insights into the performance of these algorithms.

CONCLUSION

In the literature on evolutionary optimization, many effective approaches have been proposed to hybridize GAs with Nelder-Mead simplex. More recently, researchers have begun implementing similar ideas within PSO also. A few papers on multi-objective hybrid approaches have been published. However, a formal framework to categorize all these approaches has so far been lacking. This chapter surveys the various methods and proposes a way to organize them into four distinct categories.

REFERENCES

Bersini, H. (2002). The immune and chemical crossovers. IEEE Transactions on Evolutionary Computation. 6(3): 306-313.

Chelouah, R., & Siarry, P. (2000). A continuous genetic algorithm designed for the global optimization of multimodal functions, Journal of Heuristics 6: 191-213.

Chelouah, R., & Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions, European Journal of Operational Research, 148: 335-348.

Das, S., Koduru, P., Welch, S.M., Gui, M., Cochran, M., Wareing, A., & Babin, B. (2006). Adding local search to particle swarm optimization. Proceedings, World Congress on Computational Intelligence, Vancouver, BC, Canada, 428-433.

Durand N., & Alliot, J.M. (1999). A combined Nelder-Mead Simplex and genetic algorithm. Genetic and Evolutionary Computing Conference (GECCO).

Guo G., & Shouyi, Y. (2003). Evolutionary parallel local search for function optimization. IEEE Transactions on Systems, Man and Cybernetics Part-B. 7(1): 243-258.

Hedar, A., Fukushima, M. (2003). "Minimizing multimodal functions by simplex coding genetic algorithm", Optimization Methods and Software. 18: 265-282.

Fan, S.S., Liang, Y.C., Zahara, E. (2004). "Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions", Engineering Optimization. 36(4): 401-418.

Kennedy, J., & Eberhart, R.C. (2001) Swarm Intelligence, Morgan Kaufmann, CA.

Koduru, P. Das, S., Welch, S.M., & Roe, J. (2004). Fuzzy dominance based multi-objective GA-Simplex hybrid algorithms applied to gene network models. Lecture Notes in Computer Science: Proceedings of the Genetic and Evolutionary Computing Conference, Seattle, Washington, (Eds. Kalyanmoy Deb et al.), Springer-Verlag, 3102: 356-367.

Koduru, P. Das, S., Welch, S.M., Roe, J., & Lopez-Dee, Z.P. (2005). A Co-evolutionary hybrid algorithm for multi-objective optimization of gene regulatory network models, Proceedings of the Genetic and Evolutionary Computing Conference, Washington D. C. 393-399.

Koduru, P., Das, S., & Welch, S.M. (2007). Multi-objective and hybrid PSO using ε-fuzzy dominance, Proceedings of the Genetic and Evolutionary Comput-

ing Conference, London, UK. (Eds. Dirk Thierens et al.) 853-860.

Koduru, P., Welch, S.M., & Das, S. (2007). A particle swarm optimization approach for estimating confidence regions. Proceedings of the Genetic and Evolutionary Computing Conference, London, UK. (Eds. Dirk Thierens et al.) 70-77.

Mitchell, M. (1998). An Introduction to Genetic Algorithms. MIT Press.

Nelder, J.A., & R. Mead, R. (1965). A simplex method for function minimization, Computer Journal, 7(4): 308-313.

Renders, J., & Bersini, H. (1994). Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. Proceedings of IEEE International Conference on Evolutionary Computation. 312-317.

Renders, J.M., & Flasse, S.P. (1998). Hybrid methods using genetic algorithms for global optimization, IEEE Transactions on Systems, Man and Cybernetics Part-B. 28(2): 73-91.

Robin, F., Orzati, A., Moreno, E., Homan, O.J., & Bachtold, W. (2003). Simulation and evolutionary optimization of electron-beam lithography with genetic and simplex-downhill algorithms, IEEE Transactions on Evolutionary Computation. 7(1): 69-82.

Trabia, M.B. (2004). A hybrid fuzzy simplex genetic algorithm. Journal of Mechanical Design, 126 (6): 969-97.

Yang, R., & Douglas, I., (1998). Simple genetic algorithm with local tuning: Efficient global optimizing technique. Journal of Optimization Theory and Applications. 98: 449-465.

Yen, J., Liao, J.C., Lee, B., & Randolph, D. (1998) A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method, IEEE Transactions on Systems, Man and Cybernetics Part-B, 28(2): 173-191.

Zahara, E., Fan, S.S. & Tsai, D.M. (2005). Optimal multi-thresholding using a hybrid optimization approach. Pattern Recognition Letters, 26(8): 1082-1095.

KEY TERMS

Dominance: A relationship between solutions in a multi-objective optimization problem. A solution dominates another if and only if it is equal to the latter in all the objectives and better in at least one.

Evolutionary Algorithm: A class of probabilistic algorithms that are based upon biological metaphors such as Darwinian evolution, and widely used in optimization.

Exploration: A strategy that samples the fitness landscape extensively to obtain good regions.

Exploitation: A greedy strategy that seeks to improve one or more solutions to an optimization problem to take it to a maximum in its vicinity.

Fitness: A measure to determine the goodness of a solution to an optimization problem. When a single objective is to be maximized, the fitness is either equal to the objective or a monotonically increasing function of it.

Fitness Landscape: A representation of the search space of an optimization problem that brings out the differences in the fitness of the solutions, such that those with good fitness are "higher". Optimal solutions are the maxima of the fitness landscape.

Generation: A term used in evolutionary algorithms that corresponds to an iteration of the outermost loop.

Local Search: A search algorithm to carry out exploitation.

Multi-Objective Optimization: An optimization problem involving more than a single objective function. In such a setting, it is not easy to discriminate between good and bad solutions, as a solution that is better than another in one objective may be poorer in another. Without any loss of generality, each objective function can be considered to be one involving maximization.

Population-Based Algorithm: An algorithm, which maintains an entire set of candidate solutions for an optimization problem.

Search Space: Set of all possible solutions for any given optimization problem, in which one can usually define a neighborhood of any solution.

Neural Control System for Autonomous Vehicles

Francisco García-Córdova
Polytechnic University of Cartagena (UPCT), Spain

Antonio Guerrero-González
Polytechnic University of Cartagena (UPCT), Spain

Fulgencio Marín-García
Polytechnic University of Cartagena (UPCT), Spain

INTRODUCTION

Neural networks have been used in a number of robotic applications (Das & Kar, 2006; Fierro & Lewis, 1998), including both manipulators and mobile robots. A typical approach is to use neural networks for nonlinear system modelling, including for instance the learning of forward and inverse models of a plant, noise cancellation, and other forms of nonlinear control (Fierro & Lewis, 1998).

An alternative approach is to solve a particular problem by designing a specialized neural network architecture and/or learning rule (Sutton & Barto, 1981). It is clear that biological brains, though exhibiting a certain degree of homogeneity, rely on many specialized circuits designed to solve particular problems.

We are interested in understanding how animals are able to solve complex problems such as learning to navigate in an unknown environment, with the aim of applying what is learned of biology to the control of robots (Chang & Gaudiano, 1998; Martínez-Marín, 2007; Montes-González, Santos-Reyes & Ríos-Figueroa, 2006).

In particular, this article presents a neural architecture that makes possible the integration of a kinematical adaptive neuro-controller for trajectory tracking and an obstacle avoidance adaptive neuro-controller for nonholonomic mobile robots. The kinematical adaptive neuro-controller is a real-time, unsupervised neural network that learns to control a nonholonomic mobile robot in a nonstationary environment, which is termed Self-Organization Direction Mapping Network (SODMN), and combines associative learning and Vector Associative Map (VAM) learning to generate transformations between spatial and velocity coordinates (García-Córdova, Guerrero-González & García-Marín, 2007). The transformations are learned in an unsupervised training phase, during which the robot moves as a result of randomly selected wheel velocities. The obstacle avoidance adaptive neuro-controller is a neural network that learns to control avoidance behaviours in a mobile robot based on a form of animal learning known as operant conditioning. Learning, which requires no supervision, takes place as the robot moves around a cluttered environment with obstacles. The neural network requires no knowledge of the geometry of the robot or of the quality, number, or configuration of the robot's sensors. The efficacy of the proposed neural architecture is tested experimentally by a differentially driven mobile robot.

BACKGROUND

Several heuristic approaches based on neural networks (NNs) have been proposed for identification and adaptive control of nonlinear dynamic systems (Fierro & Lewis, 1998; Pardo-Ayala & Angulo-Bahón, 2007).

In wheeled mobile robots (WMR), the trajectory-tracking problem with exponential convergence has been solved theoretically using time-varying state feedback based on the backstepping technique in (Ping & Nijmeijer, 1997; Das & Kar, 2006). Dynamic feedback linearization has been used for trajectory tracking and posture stabilization of mobile robot systems in chained form (Oriolo, Luca & Vendittelli, 2002).

The study of autonomous behaviour has become an active research area in the field of robotics. Even the simplest organisms are capable of behavioural feats unimaginable for the most sophisticated machines. When

an animal has to operate in an unknown environment it must somehow learn to predict the consequences of its own actions. Biological organisms are a clear example that this short of learning is possible in spite of what, from an engineering standpoint, seem to be insurmountable difficulties: noisy sensors, unknown kinematics and dynamics, nostationary statistics, and so on. A related form of learning is known as operant conditioning (Grossberg, 1971). Chang and Gaudiano (1998) introduce a neural network for obstacle avoidance that is based on a model of classical and operant conditioning.

Psychologists have identified classical and operant conditioning as two primary forms of learning that enables animals to acquire the causal structure of their environment. In the classical conditioning paradigm, learning occurs by repeated association of a Conditioned Stimulus (CS), which normally has no particular significance for an animal, with an Unconditioned Stimulus (UCS), which has significance for an animal and always gives rise to an Unconditioned Response (UCR). The response that comes to be elicited by the CS after classical conditioning is known as the Conditioned Response (CR) (Grossberg & Levine, 1987). Hence, classical conditioning is the putative learning process that enables animals to recognize informative stimuli in the environment.

In the case of operant conditioning, an animal learns the consequences of its actions. More specifically, the animal learns to exhibit more frequently a behaviour that has led to reward in the past, and to exhibit less frequently a behaviour that led to punishment.

In the field of neural networks research, it is often suggested that neural networks based on associative learning laws can model the mechanisms of classical conditioning, while neural networks based on reinforcement learning laws can model the mechanisms of operant conditioning (Chang & Gaudiano, 1998).

The reinforcement learning is used to acquire navigation skills for autonomous vehicles, and updates both the vehicle model and optimal behaviour at the same time (Galindo, González & Fernández-Madrigal, 2006; Lamiraux & Laumond, 2001; Galindo, Fernández-Madrigal & González, 2007).

In this article, we propose a neurobiologically inspired neural architecture to show how an organism, in this case a robot, can learn without supervision to recognize simple stimuli in its environment and to associate them with different actions.

ARCHITECTURE OF THE NEURAL CONTROL SYSTEM

Figure 1(a) illustrates our proposed neural architecture. The trajectory tracking control without obstacles is implemented by the SODMN and a neural network of biological behaviour implements the avoidance behaviour of obstacles.

Self-Organization Direction Mapping Network (SODMN)

The transformation of spatial directions to wheels angular velocities is expressed like a linear mapping and is shown in Fig. 1(b). The spatial error is computed to get a spatial direction vector (DVs). The DVs is transformed by the *direction mapping network* elements V_{ik} to corresponding motor direction vector (DVm). On the other hand, a set of tonically active inhibitory cells, which receive broad-based inputs that determine the context of a motor action, was implemented as a context field. The context field selects the V_{ik} elements based on the wheels angular velocities configuration.

A speed-control GO signal acts as a non-specific multiplicative gate and controls the movement's overall speed. The GO signal is an input from a decision centre in the brain, and starts at zero before movement and then grows smoothly to a positive value as the movement develops. During the learning, the GO signal is inactive.

Activities of cells of the DVs and DVm are represented in the neural network by quantities (S_1, S_2, ..., S_m) and (R_1, R_2, ..., R_n), respectively. The direction mapping is formed with a field of cells with activities V_{ik}. Each V_{ik} cell receives the complete set of spatial inputs $S_j, j = 1, ..., m$, but connects to only one R_i cell. The direction mapping cells ($V \in \mathbb{R}^{n \times k}$) compute a difference of activity between the spatial and motor direction vectors via feedback from DVm. During learning, this difference drives the adjustment of the weights. During performance, the difference drives DVm activity to the value encoded in the learned mapping.

A context field cell pauses when it recognizes a particular velocity state (i.e., a velocity configuration) on its inputs, and thereby disinhibits its target cells. The target cells (direction mapping cells) are completely shut off when their context cells are active (see Fig. 1(b)). Each context field cell projects to a set of

Figure 1. (a) Neural architecture for reactive and adaptive navigation of a mobile robot. (b) Self-organization direction mapping network for the trajectory tracking of a mobile robot.

(a)

(b)
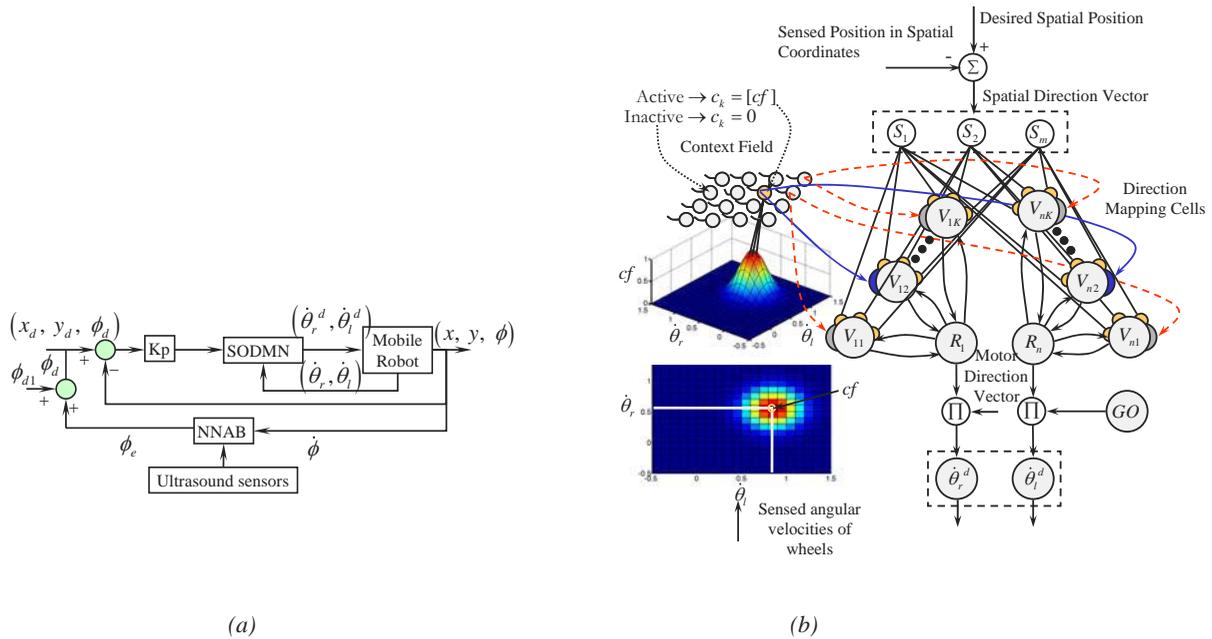
direction mapping cells, one for each velocity vector component. Each velocity vector component has a set of direction mapping cells associated with it, one for each context. A cell is "off" for a compact region of the velocity space. It is assumed for simplicity that only one context field cell turns "off" at a time. The centre context field cell is "off" when the angular velocities are in the centre region of the velocity space. The "off" context cell enables a subset of direction mapping cells through the inhibition variable c_k, while "on" context cells disable to the other subsets.

The learning is obtained by decreasing weights in proportion to the product of the presynaptic and postsynaptic activities (Gaudiano, & Grossberg, 1991). The training is done by generating random movements, and by using the resulting angular velocities and observed spatial velocities of the mobile robot as training vectors to the direction mapping network.

Neural Network for the Avoidance Behaviour (NNAB)

Grossberg proposed a model of classical and operant conditioning, which was designed to account for a

variety of behavioural data on learning in vertebrates (Grossberg, 1971; Grossberg & Levine, 1987). Our implementation is based in the Grossberg's conditioning circuit, which follows closely that of Grossberg & Levine (1987) and Chang & Gaudiano (1998), and is shown in Figure 2.

In this model the sensory cues (both CSs and UCS) are stored in Short Term Memory (STM) within the population labelled *S*, which includes competitive interactions to ensure that the most salient cues are contrast enhanced and stored in STM while less salient cues are suppressed. The population *S* is modelled as a recurrent competitive field in simplified discrete-time version, which removes the inherent noise, efficiently normalizes and contrast-enhances from the ultrasound sensors activations. In the present model, the CS nodes correspond to activation from the robot's ultrasound sensors. In the network I_i represents a sensor value which codes proximal objects with large values and distal objects with small values. The network requires no knowledge of the geometry of the mobile robot or the quality, number, or distribution of sensors over the robot's body.

Fig. 2. Neural Network for the avoidance behaviour
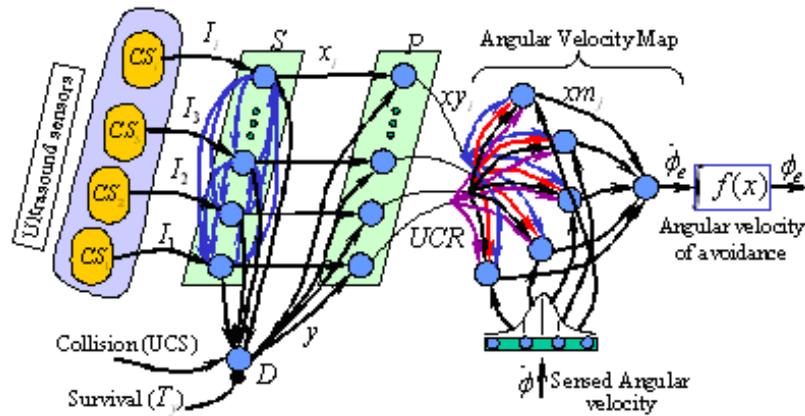
The drive node *D* corresponds to the Reward/Punishment component of operant conditioning (an animal/robot learns the consequences of its own actions). Learning can only occur when the drive node is active. Activation of drive node *D* is determined by the weighted sum of all the CS inputs, plus the UCS input, which is presumed to have a large, fixed connection strength. The drive node *D* is active when the robot collides with an obstacle. Then the unconditioned stimulus (USC) in this case corresponds to a collision detected by the mobile robot. The activation of the drive node and of the sensory nodes converges upon the population of polyvalent cells *P*. Polyvalent cells require the convergence of two types of inputs in order to become active. In particular, each polyvalent cell receives input from only one sensory node, and all polyvalent cells also receive input from the drive node *D*.

Finally, the neurons (xm_i) represent the response conditioned or unconditioned and are thus connected to the motor system. The motor population consists of nodes (i.e., neurons) encoding desired angular velocities of avoidance. When driving the robot, activation is distributed as a Gaussian centred on the desired angular velocity of avoidance. The use of a Gaussian leads to smooth transitions in angular velocity even with few nodes.

The output of the angular velocity population is decomposed by SODMN into left and right wheel angular velocities. A gain term can be used to specify the maximum possible velocity. In NNAB the proximity sensors initially do not propagate activity to the motor population because the initial weights are

small or zero. The robot is trained by allowing it to make random movements in a cluttered environment. Whenever the robot collides with an obstacle during one of these movements (or comes very close to it), the nodes corresponding to the largest (closest) proximity sensor measurements just prior to the collision will be active. Activation of the drive node *D* allows two different kinds of learning to take place: the learning that couples sensory nodes (infrared or ultrasounds) with the drive node (the collision), and the learning of the angular velocity pattern that existed just before the collision.

The first type of learning follows an associative learning law with decay. The primary purpose of this learning scheme is to ensure that learning occurs only for those CS nodes that were active within some time window prior to the collision (UCS). The second type of learning, which is also of an associative type but inhibitory in nature, is used to map the sensor activations to the angular velocity map. By using an inhibitory learning law, the polyvalent cell corresponding to each sensory node learns to generate a pattern of inhibition that matches the activity profile active at the time of collision.

Once learning has occurred, the activation of the angular velocity map is given by two components (see Figure 3). An excitatory component, which is generated directly by the sensory system, reflects the angular velocity required to reach a given target in the absence of obstacles. The second, inhibitory component, generated by the conditioning model in response to sensed obstacles, moves the robot away from the obstacles

Figure 3. Positive Gaussian distribution represents the angular velocity without obstacle and negative distribution represents activation from the conditioning circuit. The summation represents the angular velocity that will be used to drive the mobile robot.
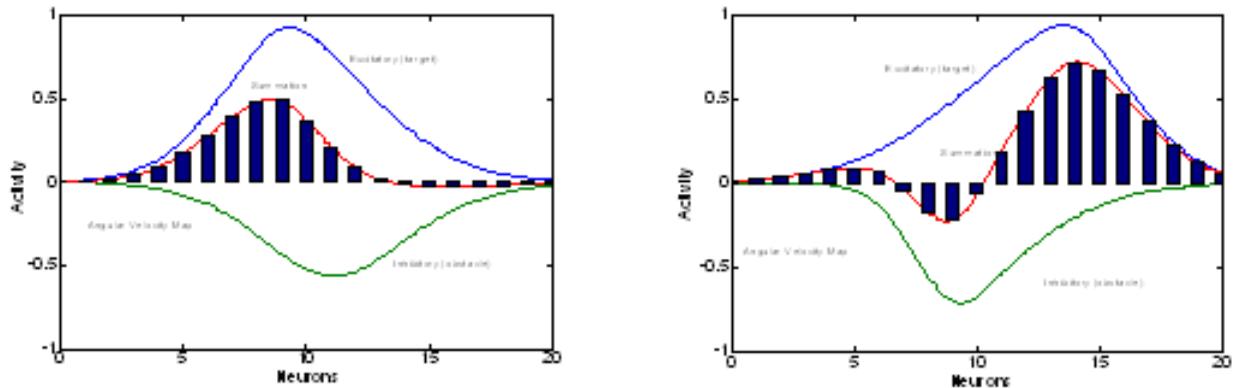
as a result of the activation of sensory signals in the conditioning circuit.

EXPERIMENTAL RESULTS

The proposed control algorithm is implemented on a mobile robot from the Polytechnic University of Cartagena (UPCT) named "CHAMAN". The platform has two driving wheels (in the rear) mounted on the same axis and two passive supporting wheels (in front) of free orientation. The two driving wheels are independently driven by two DC-motors to achieve the motion and orientation.

High-level control algorithms (SODMN and NNAB) are written in VC++ and run with a sampling time of 10 ms on a remote server (a Pentium IV processor). The lower level control layer is in charge of the execution of the high-level velocity commands. It consists of a Texas Instruments TMS320C6701 Digital Signal Processor (DSP).

Figure 4 shows approach behaviours and the tracking of a trajectory by the mobile robot with respect to the reference trajectory.

Figure 5 illustrates the mobile robot's performance in the presence of several obstacles. The mobile robot starts from the initial position labelled X and reaches a desired position. During the movements, whenever the mobile robot is approaching an obstacle, the inhibitory profile from the conditioning circuit (NNAB) changes the selected angular velocity and makes the mobile robot turn away from the obstacle.

FUTURE TRENDS

The tendency of robots' control systems is to come to understand and to imitate the way that biological systems learn and evolve to resolve complex problems in unknown environments. Simple animals (e.g.: crabs, insects, scorpions and other ones) are studied to formalize robust neural models for the robots' locomotion system. In humans, decoded neural behaviors of neural activities of the cortical system tend to be applying to robotic prosthesis for the control of movement. Neural networks and other bio-mimetic techniques with an emphasis on navigation and control are used to operate in real-time with only minimal assumptions about the robots or the environment, and that can learn, if needed, with little or no external supervision.

In this article, the proposed neural control system can be applied for underwater applications. In this case, sonar sensors will replace ultrasound sensors. The proposed neural architecture learns to carry out a reactive and adaptive navigation nonstationary environments.

Figure 4. Adaptive control by the SODMN. a) Approach behaviours. The symbol X indicates the start of the mobile robot and T_i indicates the desired reach. b) Tracking control of a desired trajectory. c) Real-time tracking performance.

(a)

(b)

(c)
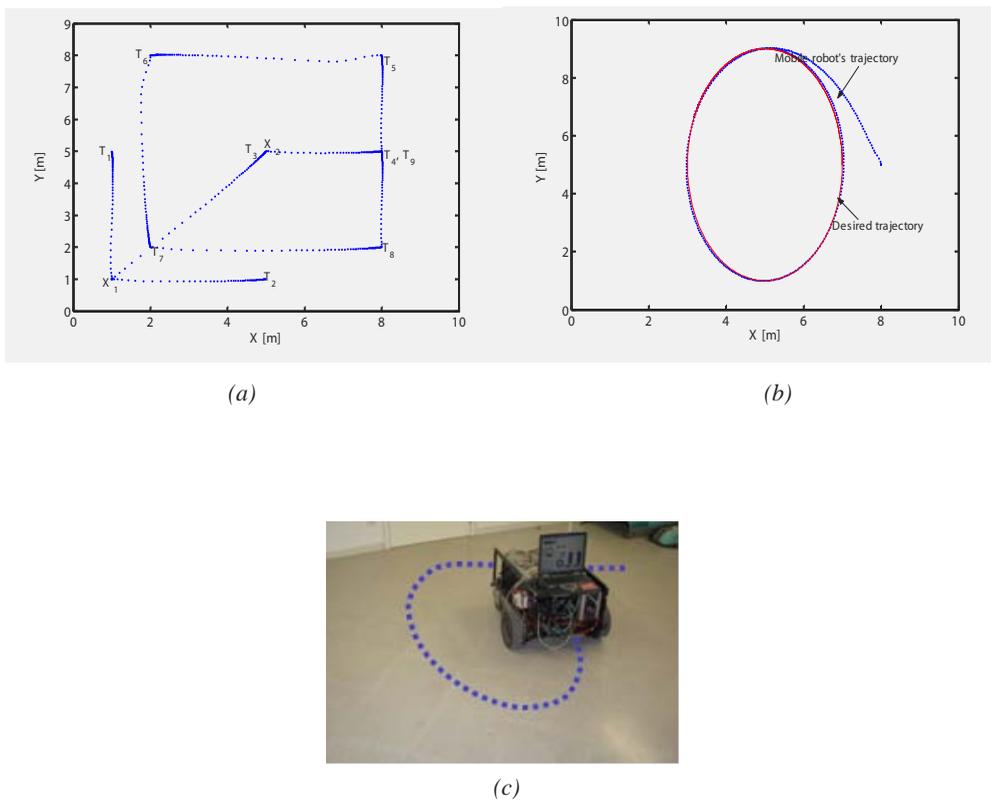
Figure 5. Trajectory followed by the mobile robot in presence of obstacles using the NNAB

(a)

(b)

N
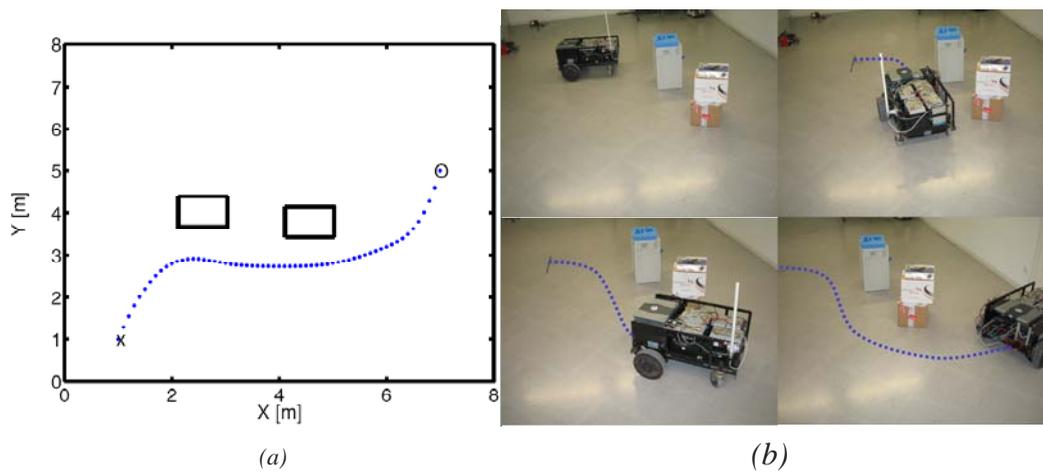
CONCLUSION

In this article, we have implemented a neural architecture for trajectory tracking and avoidance behaviours of mobile robot. A biologically inspired neural network for the spatial reaching tracking has been developed. This neural network is implemented as a kinematical adaptive neuro-controller. The SODMN uses a context field for learning the direction mapping between spatial and angular velocity coordinates. The performance of this neural network has been successfully demonstrated in experimental results with the trajectory tracking and reaching of a mobile robot. The avoidance behaviours of obstacles were implemented by a neural network that is based on a form of animal learning known as operant conditioning. A differentially driven mobile robot tested the efficacy of the proposed neural network for avoidance behaviours experimentally.

REFERENCES

Chang, C., & Gaudiano, P. (1998). Application of biological learning theories to mobile robot avoidance and approach behaviors. *J. Complex Systems*. (1), 79–114.

Das, T., & Kar, I.N. (2006). Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. *IEEE Transactions on Control SystemsTech-nology*. (14), 501–510.

Fierro, R., & Lewis, F.L. (1998). Control of a non-holonomic mobile robot using neural networks. *IEEE Trans. Neural Netw.* (9), 589–600.

Galindo, C., Fernández-Madrigal, J.A., & González, J. (2007). Towards the automatic learning of reflex modulation for mobile robot navigation. *Nature Inspired Problem-Solving Methods in Knowledge Engineering*, Mira, J., & Alvarez, J.R. (Eds.) IWINAC 2007, Part II. LNCS vol. 4528, 347-356. Springer, Heidelberg.

Galindo, C., González, J., & Fernández-Madrigal, J.A. (2006). A control architecture for human-robot integration: Application to robotic wheelchair. *IEEE Trans. On Systems, Man, and Cyb.* –Part B, 36, 1053-1068.

García-Córdova, F., Guerrero-González, A., & García-Marín, F. (2007). Design and implementation of an adaptive neuro-controller for trajectory tracking of nonholonomic wheeled mobile robots. *Nature Inspired Problem-Solving Methods in Knowledge Engineering*, Mira, J., & Alvarez, J.R. (Eds.) IWINAC 2007, Part II. LNCS vol. 4528, 459-468. Springer, Heidelberg.

Gaudiano, P., & Grossberg, S. (1991). Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories. *Neural Networks*. (4), 147–183.

Grossberg, S. (1971). On the dynamics of operant conditioning. *Journal of Theorical Biology*. (33), 225–255.

Grossberg, S., & Levine, D. (1987). Neural dynamics of attentionally moduled Pavlovian conditioning: Blocking, interstimulus interval, and secondary reinforcement. *Applied Optics*. (26), 5015–5030.

Lamiraux, F., & Laumond, J.P. (2001). Smooth motion planning for car-like vehicles. *IEEE Trans. Robot. Automat.* 17(4), 498-502.

Martínez-Marín, T. (2007). Learning autonomous behaviours for nonholonomic vehicles. *Computational and Ambient Intelligence*, Sandoval, F., Prieto, A., Cabestany, J., & Graña, M. (Eds.) IWANN 2007. LNCS vol. 4507, 839-846. Springer, Heidelberg.

Montes-González, F., Santos Reyes, J., & Ríos Figueroa, H. (2006). Integration of evolution with a robot action selection model. Gelbukh, A., & Reyes-García, C.A. (Eds.) MICAI 2006. LNCS vol. 4293, 1160-1170.

Oriolo, G., Luca, A.D., & Vendittelli, M. (2002). WMR control via dynamic feedback linearization: Design, implementation and experimental validation. *IEEE Trans. Control.Syst. Technol.* 10, 835-852.

Pardo-Ayala, D.E., & Angulo-Bahón, C. (2007). Emerging behaviors by learning joint coordination in articulated mobile robots. *Computational and Ambient Intelligence*, Sandoval, F., Prieto, A., Cabestany, J., & Graña, M. (Eds.) IWANN 2007. LNCS vol. 4507, 806-813. Springer, Heidelberg.

Ping, Z., & Nijmeijer, H. (1997). Tracking control of mobile robots: A case study in backstepping. *Automatica*. (33), 1393–1399.

Sutton, R.S., & Barto, A.G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*. (88), 135–170.

KEY TERMS

Artificial Neural Network: A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data; and are used in applications such as robotics, speech recognition, and signal processing or medical diagnosis.

Classical Conditioning: It is a form of associative learning that was first demonstrated by Ivan Pavlov. The typical procedure for inducing classical conditioning involves a type of learning in which a stimulus acquires the capacity to evoke a response that was originally evoked by another stimulus.

Conditioned Response (CR): If the conditioned stimulus and the unconditioned stimulus are repeatedly paired, eventually the two stimuli become associated and the organism begins to produce a behavioral response to the conditioned stimulus. Then, the conditioned response is the learned response to the previously neutral stimulus.

Conditioned Stimulus (CS): It is a previously neutral stimulus that, after becoming associated with the unconditioned stimulus, eventually comes to trigger a conditioned response. The neutral stimulus could be any event that does not result in an overt behavioral response from the organism under investigation.

Operant Conditioning: The term "Operant" refers to how an organism operates on the environment, and hence, operant conditioning comes from how we respond to what is presented to us in our environment. Then the operant conditioning is a form of associative learning through which an animal learns about the consequences of its behaviour.

Unconditioned Response (UR): It is the unlearned response that occurs naturally in response to the unconditioned stimulus.

Unconditioned Stimulus (UCS): Which is one that unconditionally, naturally, and automatically triggers an innate, often reflexive, response in the presence of significant stimulus. For example, when you smell one of your favourite foods, you may immediately feel very hungry. In this example, the smell of the food is the unconditioned stimulus.

Neural Network–Based Visual Data Mining for Cancer Data

Enrique Romero
Technical University of Catalonia, Spain

Julio J. Valdés
National Research Council Canada, Canada

Alan J. Barton
National Research Council Canada, Canada

INTRODUCTION

According to the World Health Organization (http://www.who.int/cancer/en), cancer is a leading cause of death worldwide. From a total of 58 million deaths in 2005, cancer accounts for 7.6 million (or 13%) of all deaths. The main types of cancer leading to overall cancer mortality are *i)* Lung (1.3 million deaths/year), ii) Stomach (almost 1 million deaths/year), *iii)* Liver (662,000 deaths/year), *iv)* Colon (655,000 deaths/year) and *v)* Breast (502,000 deaths/year). Among men the most frequent cancer types worldwide are (in order of number of global deaths): lung, stomach, liver, colorectal, oesophagus and prostate, while among women (in order of number of global deaths) they are: breast, lung, stomach, colorectal and cervical.

Technological advancements in recent years are enabling the collection of large amounts of cancer related data. In particular, in the field of Bioinformatics, high-throughput microarray gene experiments are possible, leading to an information explosion. This requires the development of data mining procedures that speed up the process of scientific discovery, and the in-depth understanding of the internal structure of the data. This is crucial for the non-trivial process of identifying valid, novel, potentially useful, and ultimately *understandable patterns* in data (Fayyad, Piatesky-Shapiro & Smyth, 1996). Researchers need to *understand* their data rapidly and with greater ease. In general, objects under study are described in terms of collections of *heterogeneous* properties. It is typical for medical data to be composed of properties represented by nominal, ordinal or real-valued variables (scalar), as well as by others of a more complex nature, like images, time-series, etc. In addition, the information

comes with different degrees of precision, uncertainty and information completeness (missing data is quite common).

Classical data mining and analysis methods are sometimes difficult to use, the output of many procedures may be large and time consuming to analyze, and often their interpretation requires special expertise. Moreover, some methods are based on assumptions about the data which limit their application, specially for the purpose of exploration, comparison, hypothesis formation, etc, typical of the first stages of scientific investigation. This makes graphical representation directly appealing. Humans perceive most of the information through vision, in large quantities and at very high input rates. The human brain is extremely well qualified for the fast understanding of complex visual patterns, and still outperforms the computer. Several reasons make Virtual Reality (VR) a suitable paradigm: *i)* it is *flexible* (it allows the choice of different representation models to better suit human perception preferences), *ii)* allows *immersion* (the user can navigate inside the data, and interact with the objects in the world), *iii)* creates a *living* experience (the user is not merely a passive observer, but an actor in the world) and *iv)* VR is *broad and deep* (the user may see the VR world as a whole, and/or concentrate on specific details of the world). Of no less importance is the fact that in order to interact with a virtual world, only minimal skills are required.

Visualization techniques may be very useful for medical decisión support in the oncology area. In this paper unsupervised neural networks are used for constructing VR spaces for visual data mining of gene expression cancer data. Three datasets are used in the paper, representative of three of the most important

types of cancer in modern medicine: liver, stomach and lung. The data sets are composed of samples from normal and tumor tissues, described in terms of tens of thousands of variables, which are the corresponding gene expression intensities measured in microarray experiments. Despite the very high dimensionality of the studied patterns, high quality visual representations in the form of structure-preserving VR spaces are obtained using SAMANN neural networks, which enables the differentiation of cancerous and noncancerous tissues. The same networks could be used as nonlinear feature generators in a preprocessing step for other data mining procedures.

NEURAL NETWORKS FOR THE CONSTRUCTION OF VIRTUAL REALITY SPACES

VR spaces for the visual representation of information systems (Pawlak, 1991) and relational structures were introduced in (Valdés, 2002a) (Valdés, 2003). A *VR space* is a tuple $\Omega = <\underline{O}, G, B, R^m, g_0, l, g_r, b, r>$, where $\underline{O}$ is a relational structure ($\underline{O} = <O, \Gamma^v>$), O is a finite set of objects, and Γ^v is a set of relations); G is a non-empty set of *geometries* representing the different objects and relations; B is a non-empty set of *behaviors* of the objects in the virtual world; $R^m \subseteq \Re^m$ is a *metric space* of dimension m (Euclidean or not) which will be the actual VR geometric space. The other elements are mappings: $g_0 : O \rightarrow G$, $l : O \rightarrow R^m$, $g_r : \Gamma^v \rightarrow G$ and $b : O \rightarrow B$.

The typical *desiderata* for the visual representation of data and knowledge can be formulated in terms of minimizing information loss, maximizing structure preservation, maximizing class separability, or their combination, which leads to single or multi-objective optimization problems. In many cases, these concepts can be expressed deterministically using continuous functions with well defined partial derivatives. This is the realm of classical optimization where there is a plethora of methods with well known properties. In the case of heterogeneous information the situation is more complex and other techniques are required (Valdés, 2002b) (Valdés, 2004) (Valdés & Barton, 2005). In the unsupervised case, the function f mapping the original space to the VR (geometric) space R^m can be constructed as to maximize some metric/non-metric structure preservation criteria as is typical in multidimensional

scaling (Borg & Lingoes, 1987) or minimize some error measure of information loss (Sammon, 1969). A typical error measure is:

$$Sammon\ Error = \frac{1}{\sum\limits_{i<j}\delta_{ij}} \frac{\sum\limits_{i<j}(\delta_{ij}-\xi_{ij})^2}{\delta_{ij}}$$

where δ_{ij} is a dissimilarity measure between two objects i, j in the original space, and ξ_{ij} is another dissimilarity measure defined on objects i, j in the VR space (the images of i, j under f). Typical dissimilarity measures for δ_{ij} are the Euclidean distance or the dissimilarity based on Gower's similarity coefficient (Gower, 1971). The Euclidean distance is the usual measure for ξ_{ij} in the VR space.

Usually, the mappings f obtained using approaches of this kind are *implicit* because the images of the objects in the new space are computed directly. However, a functional representation of f is highly desirable, specially in cases where more samples are expected *a posteriori* and need to be placed within the space. With an implicit representation, the space has to be computed every time that a new sample is added to the set, whereas with an explicit representation, the mapping can be computed directly. As long as the incoming objects can be considered as belonging to the same population of samples used for constructing the mapping function, the space does not need to be recomputed. Neural networks are natural candidates for constructing explicit representations due to their general universal approximation property. If proper training methods are used, neural networks can learn structure preserving mappings of high dimensional samples into lower dimensional spaces suitable for visualization (2D, 3D). If visualization is not a requirement, spaces of smaller dimension than the original can be used as new features for noise reduction or other data mining methods. Such an example is the SAMANN network. This is a feedforward network and its architecture consists of an input layer with as many neurons as descriptor attributes, an output layer with as many neurons as the dimension of the VR space and one or more hidden layers. The classical way of training the SAMANN network is described in (Mao & Jain, 1995). It consists of a gradient descent method where the derivatives of the Sammon error are computed in a similar way to the classical backpropagation algorithm. Different from the backpropagation algorithm,

the training is unsupervised and the weights can only be updated alter a pair of examples are presented to the network.

CANCER DATA SETS DESCRIPTION

Three microarray gene expression cancer databases were selected. They are representative of some of the leading causes of cancer death in the world and share the typical features of these kind of data: a small number of samples (in the order of tens), described in terms of a very large number of attributes (in the order of tens of thousands).

Liver Cancer Data

We used the same data as in (Lam, Wu, Vega, Miller, Spitsbergen, Tong, Zhan, Govindarajan, Lee, Mathavan, Murthy, Buhler, Liu & Gong, 2006), where zebrafish liver tumors were analyzed and compared with human liver tumors. The database (http://www.ncbi.nlm.nih.gov/projects/geo/gds/gds_browse.cgi?gds=2220) contains 20 samples (10 normal, 10 tumor), with 16,512 attributes. First, liver tumors in zebrafish were generated by treating them with carcinogens. Then, the expression profiles of zebrafish liver tumors were compared with those of zebrafish normal liver tissues using a Wilcoxon rank-sum test. As a result of this comparison, a zebrafish liver tumor differentially expressed gene set consisting of 2,315 gene features was obtained. This data set was used for comparison with human tumors. The results suggest that the molecular similarities between zebrafish and human liver tumors are greater than the molecular similarities between other types of tumors (stomach, lung and prostate).

Stomach Cancer Data

We used the same data as in (Hippo, Taniguchi, Tsutsumi, Machida, Chong, Fukayama, Kodama & Aburatani, 2002), where a study of genes that are differentially expressed in cancerous and noncancerous human gastric tissues was performed. The database (http://www.ncbi.nlm.nih.gov/projects/geo/gds/gds_browse.cgi?gds=1210) contains 30 samples (22 tumor, 8 normal) that were analyzed by oligonucleotide microarray, obtaining the expression profiles for 6,936 genes (7,129 attributes). Using the 6,272

genes that passed a prefilter procedure, cancerous and noncancerous tissues were successfully distinguished with a two-dimensional hierarchical clustering using Pearson's correlation. However, the clustering results used most of the genes on the array. To identify the genes that were differentially expressed between cancer and noncancerous tissues, a Mann-Whitney's U test was applied to the data. As a result of this analysis, 162 and 129 genes showed a higher expression in cancerous and noncancerous tissues, respectively. In addition, several genes associated with lymph node metastasis and histological classification (intestinal, diffuse) were identified.

Lung Cancer Data

We used the same data as in (Spira, Beane, Pinto-Plata, Kadar, Liu, Shah, Celli & Brody, 2004), where gene expressions were compared in for severely emphysematous lung tissue (from smokers at lung volume reduction surgery) and normal or mildly emphysematous lung tissue (from smokers undergoing resection of pulmonary nodules). The database (http://www.ncbi.nlm.nih.gov/projects/geo/gds/gds_browse.cgi?gds=737) contains 30 samples (18 severe emphysema, 12 mild or no emphysema), with 22,283 attributes. Genes with large detection P-values were filtered out, leading to a data set with 9,336 genes, that were used for subsequent analysis. Nine classification algorithms were used to identify a group of genes whose expression in the lung distinguished severe emphysema from mild or no emphysema. First, model selection was performed for every algorithm by leave-one-out cross-validation, and the gene list corresponding to the best model was saved. The genes reported by at least four classification algorithms (102 genes) were chosen for further analysis. With these genes, a two-dimensional hierarchical clustering using Pearson's correlation was performed that distinguished between severe emphysema and mild or no emphysema. Other genes were also identified that may be causally involved in the pathogenesis of the emphysema.

EXPERIMENTAL SETTINGS

Data Preprocessing

For stomach and lung data, each gene was scaled to mean zero and standard deviation one (original data

were not normalized). For liver data, no transformation was performed (original data were $\log_2$ ratios).

Model Training

For every data set, SAMANN networks were constructed to map the original data to a 3D VR space. The Euclidean distance was the dissimilarity measure used for both the original and the VR spaces. The activation functions used were sinusoidal for the first hidden layer and hyperbolic tangent for the rest. A collection of models was obtained by varying some of the network controlling parameters: number of units in the first hidden layer (two different values), weights ranges in the first hidden layer (three different values), learning rates (three different values), momentum (three different values), number of pairs presented to the network at every iteration (three different values), number of iterations (three different values) and random seeds (four different values), for a total of 1,944 SAMANN networks for every data set.

Computing Environment

All of the experiments were conducted on a Condor pool (http://www.cs.wisc.edu/condor) located at the Institute for Information Technology, National Research Council Canada.

RESULTS

For every data set, we constructed the histograms of the Sammon error for the obtained networks. All of the empirical distributions were positively skewed (with the mode on the lower error side), which is a good behavior. In addition, the general error ranges were small. In table 1 some statistics of the experiments are presented: minimum, maximum, mean and Standard deviation for the best (i.e., with smallest Sammon error) 1,000 networks.

Clearly, it is impossible to represent a VR space on printed media (navigation, interaction, and world changes are all lost). Therefore, very simple geometries were used for objects and only snapshots of the virtual worlds are presented. Figures 1, 2 and 3 show the VR spaces corresponding to the best networks for the liver, stomach and lung cancer data sets respectively. Although the mapping was generated from an unsupervised perspective (i.e., without using the class labels), objects from different classes are differently represented in the VR space for comparison purposes. Transparent membranes wrap the corresponding classes, so that the degree of class overlapping can be easily seen. In addition, it allows to look for particular samples with ambiguous diagnostic decisions.

The low values of the Sammon error indicate that the spaces preserved most of the distance structure of the data, therefore, giving a good idea about the distribution in the original spaces. The three virtual spaces are clearly polarized with two distribution modes, each one corresponding to a different class. Note, however, that classes are more clearly differentiated for the liver and stomach data sets than for the lung data set, where a certain level of overlapping exists. The reason for this may be that mild and no emphysema were considered members of the same class (see above).

The advantage of using SAMANN networks is that, since the mapping *f* between the original and the virtual space is *explicit*, a new sample can be easily transformed and visualized in the virtual space. Since the distance between any two objects is an indication of their dissimilarity, the new point is more likely to belong to the same class of its nearest neighbors. In the same way, outliers can be readily identified, although they may result from the space deformation inevitably introduced by the dimensionality reduction.

Table 1. Statistics of the best 1,000 SAMANN networks obtained

	Sammon Error			
Data Set	**Minimum**	**Maximum**	**Mean**	**Std.Dev.**
Liver Cancer	0.039905	0.055640	0.049857	0.003621
Stomach Cancer	0.062950	0.077452	0.072862	0.003346
Luna Cancer	0.079242	0.107842	0.094693	0.006978

CONCLUSION

High quality virtual reality spaces for visual data mining of typical examples of gene expression cancer data were obtained using unsupervised structure-preserving neural networks in a distributed computing data mining (grid) environment. These results show that a few nonlinear features can effectively capture the

similarity structure of the data and also provide a good differentiation between the cancer and normal classes. A similar study can be found in (Valdés, Romero & González, 2007).

However, in cases where the descriptor attributes are not directly related to class structure or where there are many noisy or irrelevant attributes the situation may not be as clear. In these cases, feature subset selection and other data mining procedures could be considered in a preprocessing stage.

Figure 1. VR space of the liver cancer data set (Sammon error = 0.039905, best out of 1,944 experiments). Dark spheres: normal, Light spheres: cancerous samples.
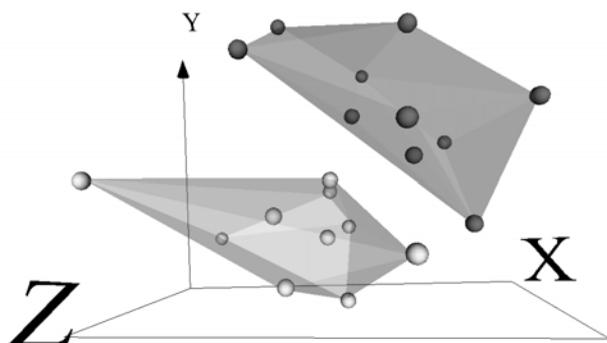
Figure 2. VR space of the stomach cancer data set (Sammon error = 0.062950, best out of 1,944 experiments). Dark spheres: normal, Light spheres: cancerous samples.
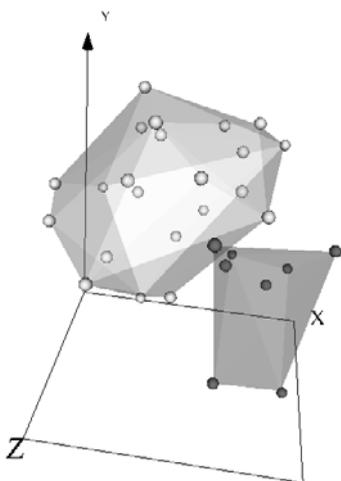
Figure 3. VR space of the lung cancer data set (Sammon error = 0.079242, best out of 1,944 experiments). Dark spheres: severe emphysema, Light spheres: mild or no emphysema. The boundary between the classes in the VR space seem to be a low curvature surface.
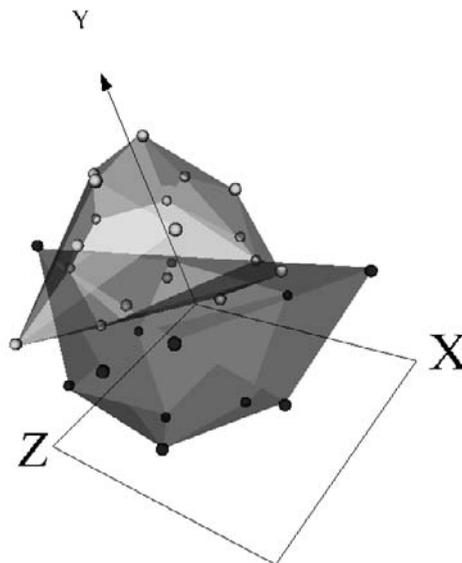
REFERENCES

Borg, I. & Lingoes, J. (1987). *Multidimensional Similarity Structure Analysis*. Springer-Verlag.

Fayyad, U., Piatesky-Shapiro, G. & Smyth (1996). From Data Mining to Knowledge Discovery. *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, et al. editors, 1-34, AAAI Press.

Gower, J.C. (1971). A General Coefficient of Similarity and Some of Its Properties. *Biometrics* 1, 857-871.

Hippo, Y., Taniguchi, H., Tsutsumi, S., Machida, N., Chong, J.M., Fukayama, M., Kodama, T. & Aburatani, H. (2002). Global Gene Expression Analysis of Gastric Cancer by Oligonucleotide Microarrays. *Cancer Research* 62 (1), 233-240.

Lam, S.H., Wu, Y.L., Vega, V.B., Miller, L.D., Spitsbergen, J., Tong, Y., Zhan, H., Govindarajan, K.R., Lee, S., Mathavan, S., Murthy, K.R.K., Buhler, D.R., Liu, E.T. & Gong, Z. (2006). Conservation of Gene Expression Signatures between Zebrafish and Human Tumors and Tumor Progression. *Nature Biotechnology* 24 (1), 73-75.

Mao, J. & Jain, A.K. (1995). Artificial Neural Networks for Feature Extraction and Multivariate Data Projection. *IEEE Transactions on Neural Networks* 6, 296-317.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers.

Sammon, J.W. (1969). A Non-linear Mapping for Data Structure Analysis. *IEEE Transactions on Computers* C-18, 401-408.

Spira, A., Beane, J., Pinto-Plata, V., Kadar, A., Liu, G., Shah, V., Celli, B. & Brody, J.S. (2004). Gene Expression Profiling of Human Lung Tissue from Smokers with Severe Emphysema. *American Journal of Respiratory Cell and Molecular Biology* 31, 601-610.

Valdés, J.J. (2002a). Virtual Reality Representation of Relational Systems and Decision Rules: An Exploratory Tool for Understanding Data Structure. *Theory and Application of Relational Structures as Knowledge Instruments*, P. Hajek editor, Meeting of the COST action 274.

Valdés, J.J. (2002b). Similarity-based Heterogeneous Neurons in the Context of General Observational Models. *Neural Network World* 12 (5), 499-508.

Valdés, J.J. (2003). Virtual Reality Representation of Information Systems and Decision Rules: An Exploratory Tool for Understanding Data and Knowledge. *International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing* (LNAI 2639), 615-618.

Valdés, J.J. (2004). Building Virtual Reality Spaces for Visual Data Mining with Hybrid Evolutionary-classical Optimization: Application to Microarray Gene Expression Data. *IASTED International Joint Conference on Artificial Intelligence and Soft Computing*, 161-166.

Valdés, J.J. & Barton, A. (2005). Virtual Reality Visual Data Mining with Nonlinear Discriminant Neural Networks: Application to Leukemia and Alzheimer Gene Expresión Data. *International Joint Conference on Neural Networks*, 2475-2480.

Valdés, J.J., Romero, E. & González, R. (2007). Data and Knowledge Visualization with Virtual Reality Spaces, Neural Networks and Rough Sets: Application to Geophysical Prospecting. *International Joint Conference on Neural Networks*, 1060-1065.

KEY TERMS

Artificial Neural Networks: Interconnected group of simple units (neurons) that, as a function of the connections between the units and the parameters, can compute complex behaviors and find nonlinear relationships in data. They are used in applications such as robotics, signal processing, or medical diagnosis.

Backpropagation Algorithm: Algorithm to compute the gradient with respect to the weights, used for the training of some types of artificial neural networks. It was first described by P. Werbos in 1974, and further developed by D.E. Rumelhart, G.E. Hinton and R.J. Williams in 1986.

Condor: Specialized workload management system for computer-intensive jobs in a distributed computing environment, developed at the university of Wisconsin-Madison (http://www.cs.wisc.edu/condor). It provides a job queuing mechanism, resource

monitoring and management, scheduling policy, and priority scheme.

Data Mining: Nontrivial extraction of implicit, previously unknown and potentially useful information from data. Typically, analytical methods and tools are applied to data with the aim of identifying patterns, relationships or obtaining databases for tasks such as classification, prediction, estimation or clustering.

Gene Expression: Process by which the inheritable information which comprises a gene, such as the DNA sequence, is made manifest as a physical and biologically functional gene product, such as protein or RNA.

SAMANN Neural Networks: Unsupervised feedforward neural networks for data projection. The classical way of training SAMANN networks was described by J. Mao and A.K. Jain in 1995. It consists of a gradient descent method where the derivatives of the Sammon error are computed in a similar way to the backpropagation algorithm.

Sammon Error: Error function to maximize structure preservation in projected data. It is defined as

$$\frac{1}{\sum\limits_{i<j} \delta_{ij}} \frac{\sum\limits_{i<j} (\delta_{ij} - \xi_{ij})^2}{\delta_{ij}},$$

where δ_{ij} and ξ_{ij} are dissimilarity measures between two objects i, j in the original and projected space, respectively.

Virtual Reality: Technology which allows the user to interact with a computer-simulated environment. Most current virtual reality environments are mainly visual experiences, displayed either on a computer screen or through special stereoscopic displays. Some advanced haptic systems include tactile information.

N

Neural Network–Based Process Analysis in Sport

Juergen Perl
University of Mainz, Germany

INTRODUCTION

Processes in sport like motions or games are influenced by communication, interaction, adaptation, and spontaneous decisions. Therefore, on the one hand, those processes are often fuzzy and unpredictable and so have not extensively been dealt with, yet. On the other hand, most of those processes structurally are roughly determined by intention, rules, and context conditions and so can be classified by means of information patterns deduced from data models of the processes.

Self organizing neural networks of type Kohonen Feature Map (KFM) help for classifying information patterns – either by mapping whole processes to corresponding neurons (see Perl & Lames, 2000; McGarry & Perl, 2004) or by mapping process steps to neurons, which then can be connected by trajectories that can be taken as process patterns for further analyses (see examples below). In any case, the dimension of the original data (i.e. the number of contained attributes) is reduced to the dimension of the representing neuron (normally 2 or 3), which makes it much easier to deal with.

Additionally, extensions of the KFM-approach are introduced, which are able to flexibly adjust the net to dynamically changing training situations. Moreover, those extensions allow for simulating adaptation processes like learning or tactical behaviour.

Finally, a current project is introduced, where tactical processes in soccer are analysed under the aspect of simulation-based optimization.

BACKGROUND

A major problem in analysing complex processes in sport like motions or games often is the reduction of available data to useful information. Two examples shall make plain what the particular problems in sport are:

In Motor Analysis, a lot of data regarding positions, angels, speed, or acceleration of articulations can be recorded automatically by means of markers and high speed digital cameras. The problem is that those recorded data show a high degree of redundancy and inherent correlation: A leg consisting of thigh, lower leg, foot, and the articulations hip joint, knee, and ankle obviously has only a comparably small range of possible movements due to natural restrictions. Therefore the quota of characteristic motion data is comparably small as well. Classification can help for deducing that relevant information from recorded data by mapping them to representative types or patterns.

In Game Analysis, during the last about 5 years an increasing number of approaches have been developed which enable for automatic recording of position data. Based on the video time precision of 25 frames per second, 9.315.000 x-y-z-coordinate data from 22 players and the ball can be taken from a 90-minutes soccer game. Obviously, the amount of data has to be reduced and to be focused to the major tactical patterns of the teams. Similar to what coaches are doing, the collection of players' positions can be reduced to constellations of tactical groups which interact like super-players and therefore enable for a computer-aided game analysis based on pattern analysis.

As is demonstrated in the following, neural network-based pattern analysis can support the handling of those problems.

MAIN FOCUS OF THE CHAPTER

Artificial Neural Networks

Current developments in the fields of Soft Computing and/or Computational Intelligence demonstrate how information patterns can be taken from data collections by means of fuzziness, similarity and learning, which

the approach of Artificial Neural Networks gives an impressive example for. In particular self organizing neural networks of type KFM (Kohonen Feature Map) play an important role in aggregating input data to clusters or types by means of a self organized similarity analysis (Kohonen, 1995).

Net-Based Process Analysis

Processes can be mapped to attribute vectors – in a game, for example, by recording the positions of the players – which then can be learned by neurons. There is, of course, a certain loss of precision if replacing an attribute vector by a representing neuron, the entry of which is similar but normally not identical to that attribute vector. Nevertheless, there are two major advantages of the way a KFM maps input data to corresponding neurons:

1. The number of objects is dramatically reduced if using the representing neurons instead of the original attribute vectors: a 2-dimensional 20×20-neuron-matrix contains 400 neurons, while a 10-dimensional vector space with only 10 different values per attribute already contains $10^{10} = 10.000.000.000$ vectors.
2. The dimension of input data is reduced to the dimension of the network (i.e. normally 2 or at most 3). This for example enables for mapping time-series of high-dimensional attribute vectors to trajectories of neurons that can easily be presented graphically.

There are three ways of gaining information from data by means of Artificial Neural Networks of KFM-type:

1. Neurons represent classes of similar data and so define types of information patterns.
2. Clusters of neurons represent time-static classes of similar information patterns and so build structures of information patterns.
3. Trajectories of neurons represent time-dynamic sequences of information patterns and so build 2-dimensional mappings of time-dependent processes. Moreover, trajectories themselves build patterns and therefore can be input to a network for classifying their similarities – which is extremely helpful not least in motor analysis or in game analysis.

There are a large number of successful applications that demonstrate how those neural networks can be used for that pattern analysis (see Perl & Dauscher, 2006).

Example "Gait Analysis": Reduction of Redundancy and Dimensionality

In gait analysis, data from articulations like for example hip-joint, knee and ankle can automatically be recorded using markers and so build a time series of n-dimensional attribute vectors which can be trained to a net. The result is that each of those n-dimensional vectors is mapped to a 2-dimensional neuron of the net – i.e. the dimension is reduced from n to 2. Corresponding to the original time series the neurons can be connected by a

Figure 1. Two trajectories of the same gait process, using 20 attribute values (left) and 10 attribute values (right), respectively. The high degree of similarity suggests that the missing 10 values are redundant and can be neglected.
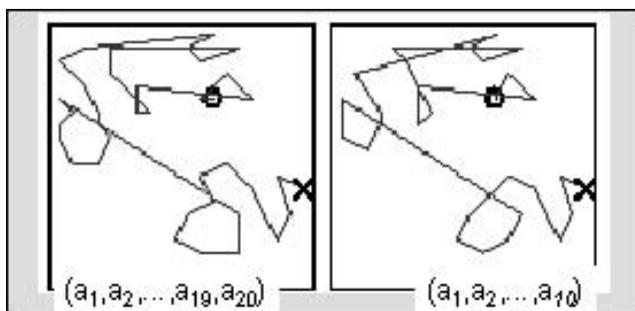
trajectory, which represents the original n-dimensional process through a 2-dimensional trajectory – therefore enabling for a much easier similarity analysis (Perl, 2004; Schöllhorn, 2004). Moreover, net-based analysis shows that, by avoiding redundancy, also the dimension of the original data can be reduced without loosing relevant information (see Figure 1).

Example "Ergometer Rowing": Inter- and Intra-Individual Process-Analysis

With the same approach that was used for gait analysis, the process of rowing was analyzed under the aspect of inter-individual similarity and intra-individual stability. Obviously, there is a great similarity on the set of all trajectories (see Figure 2).

However, the trajectories of rower A are perfectly similar to each other – demonstrating a high stability – while those of rower B are not as much. The experience with rowing pattern is that net-based analysis of rowing trajectories is very sensitive and helps for detecting even small instabilities which otherwise could not have been detected from video frames or original time series of data vectors (see Perl & Baca, 2003).

Example "Tactics in Games": Constellation Analysis

In a more complex way, trajectories can improve the transparency of the tactical behaviour of players or even a team (net-based volleyball analysis: Jäger, Perl & Schöllhorn, 2007). A collection of player positions

– i.e. a constellation – can be represented by a vector of position coordinates, which then a net can be trained with. Figure 3 shows two exemplars of the same trained volleyball-net, with small squares representing activated constellations and marked areas representing major constellation types. Obviously, the teams represented by the left and the right net activate quite different types of constellation. Moreover, the moves between the constellations – i.e. the edges and/or trajectories – are quite different, too: The left team moves between the areas, while the right team more or less selects an area and then adjusts its constellation.

In a game like volleyball – i.e. with separated teams – it is comparably easy to deduce tactical ideas from those trajectory patterns. Some first result could be taken from handball too, where net-based analysis was helpful for detecting successful offence processes (net-based handball analysis: Pfeiffer & Perl, 2006; net-based soccer analysis: Lees et al., 2003; Leser, 2006). Based on those results, currently a project is run which deals with simulation-based tactics-optimisation in soccer. First results are encouraging. They were shown as video-representation at the famous Documenta-exhibition on fine arts, 2007 in Kassel/Germany.

Dynamic Extensions of KFM-Type Neural Networks

Self organizing maps of KFM-type are very helpful for analyzing dynamic processes. They fail, however, if learning or other process dynamics are parts of the processes to be trained. This is due to the fact that the

Figure 2. Trajectories of the rowing process of two rowers A and B, one stroke per graphic
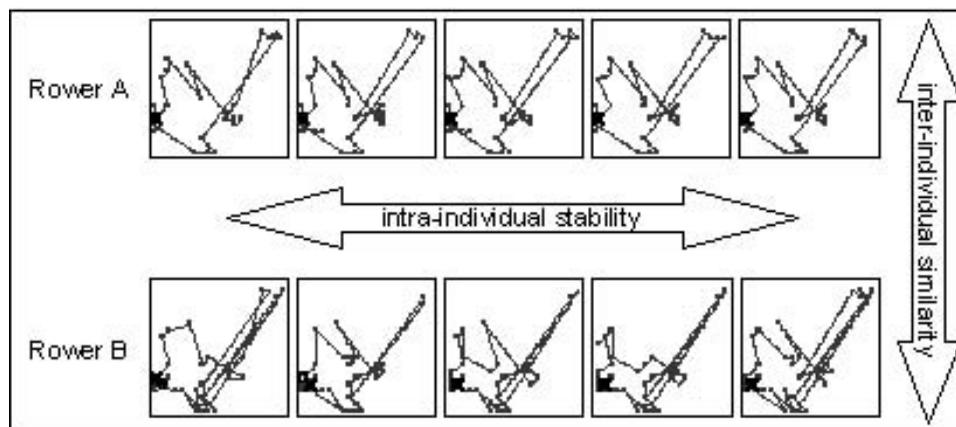
Figure 3. Two examples of a net trained with constellations, where the marked squares represent frequent constellations and the marked areas represent major types of constellations.
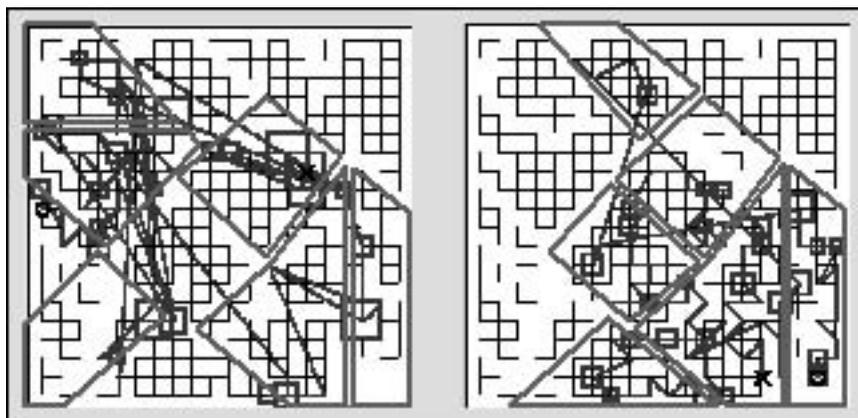
learning procedure of a KFM is externally controlled, resulting in a network that works like a tool, without being able to change with or adapt to changing process types or contexts.

One successful approach that improves the dynamics of the learning process is that of the Dynamically Controlled Network (DyCoN: Perl, 2002 a/b), which is a KFM-derivate that is able to learn continuously. The idea is that each neuron contains an individual adaptive learning model based on the Performance Potential Metamodel (PerPot: Perl, 2002 a; learning strategies: Perl & Weber, 2004).

While DyCoN helps for analysing dynamic learning processes, a different type of neural network is necessary for simulating those learning processes – in order to eventually schedule and optimize those processes individually. One important point was to dynamically adapt the capacity of the network to the requirements of the learning process. This was done by integrating the concept of Growing Neural Gas (GNG: Fritzke, 1997), where, briefly spoken, the number and positions of neurons vary time-dependently with the changing information flow from training, this way adapting the network size and topology to the training amount and content. The result is the Dynamically Controlled Neural Gas (DyCoNG) the concept of which completes the combination of DyCoN and GNG by specific „quality neurons" that reflect the information theoretical quality of information and therefore can measure the originality of a recorded activity (Perl et al., 2006). Based on the assumption that there is a strong correspondence

between the „quality" of a neuron and the originality of the represented type of activity, the network's reaction on an input-stimulus (i.e. generating a new connected/not connected quality neuron or not) indicates an evaluation of the originality of the corresponding activity. According to the two tasks „analysis of creativity learning" and „simulation of creativity learning", two major results could be obtained:

The DyCoN-model was used for analyzing the learning profiles, which were fed into as patterns and then recognized as members of clusters respectively types of learning behaviour. It was remarkable that the net could detect a number of significantly different types of learning behaviour – which in practice is useful for individually adjust the training to the athletes (Perl et al., 2006).

The DyCoNG-model was used for learning profile simulation, with the original activity- and rating-data as input and learning profiles as output. The learning profiles resulting from DyCoNG-training could also be separated into types which qualitatively correspond to those from DyCoN-analysis. This at least gives an idea of how to manage the above mentioned individual adaptation by means of net-based simulation.

In a first approach net-based originality analysis has successfully been used in case of handball: In a case study dealing with data from the Handball World Championship 2007 in Germany, offence activities of high originality could be detected with a remarkable high accordance to experts' evaluation. Moreover, a degree of originality per team and game could be mea-

sured, resulting in team-specific originality profiles that characterize increasing and decreasing playing qualities during the tournament. Currently, a similar project is run with soccer, where in a first attempt the final of the World Championship 2006 is analyzed.

FUTURE TRENDS

The two major ideas for planned future work are to expand net-based simulation of originality to associative behaviour and to analyse the effects of virtually generated "creative" activities in simulated games.

Net-Based Simulation of Associative Behaviour

In a simplified way behaviour can be understood as recognizing the behavioural context like environment or situation followed by a context-oriented selection of a best fitting activity. In case of convergent behaviour this selection is more or less rule-based and determined. In case of divergent or creative behaviour the selection has a certain undetermined degree of freedom – i.e. spontaneous „jumps" are possible from a first priority activity to associated ones. Mapped to neural networks, where activities can be thought to be connected to neurons, this means a „jump" from the input-corresponding neuron to a different one – located either in a neighboured cluster or as an isolated quality neuron

Figure 4. Net with clusters (marked by slim lines), associative „jumps" between clusters (bold dotted lines), and generated quality neuron (bold line)
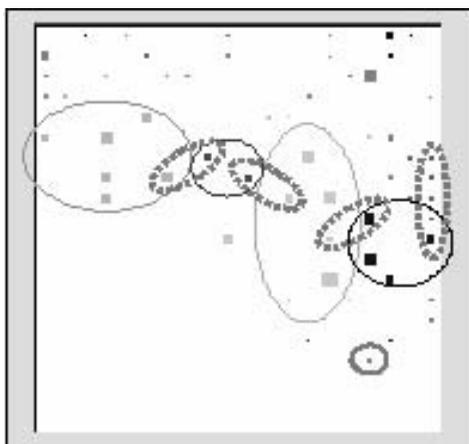
(see Figure 4). Such an associative network could help for an improved simulation of „creative" behaviour, based on a specific creativity potential that describes frequency, maximal distance, or neuron similarity of those associative jumps.

Improvement of Tactical Process Patterns

The idea of optimizing strategies by means of simulation was developed in the early 1980ies for games like tennis or badminton, where the player's abilities and tactics in a simplified way can be characterized by two matrices: The action-depending transfer of situations can be measured by a transfer frequency matrix, while the situation-depending success of actions can be measured by an action success matrix. Based on those two matrices of both the players, a game can be simulated stochastically regarding its main process structures. Moreover, modifying the entries of the matrices – i.e. changing tactical aspects or technical skills – can help for improving tactical patterns by means of simulation.

Although soccer is much more complex then tennis or badminton, the same idea can be used if the complexity is reduced by introducing „super-players" as we do in a current project: Groups of players, e.g. representing offence or defence, are combined to corresponding data objects, which are characterized by constellations of player positions. The interactions of the single players then are reduced to the interactions of the constellations or super-player, which makes it much easier to map the processes to networks for tactical analysis. The intended aim is to derive those characteristic matrices as well as information about creativity from the network in order to simulate games and improve tactical process patterns: As is indicated in Figure 5, a recorded original activity (white dot on the net) could be replaced by a apparently better or more creative one (white circle above the white dot), which in the simulation changes the regarding constellation and the resulting process and its success.

CONCLUSION

Net-based analysis of processes in sport is a difficult and challenging task because of the fuzziness and the indeterminism of athletes' behaviour and interaction.

Figure 5. Steps of net-based analysis and simulation of games like soccer: Replacing players by positions and positions by constellations; analysing constellations by means of networks; simulative modification of tactical patterns; analysing simulated games in order to improve tactical and creative behaviour.

N
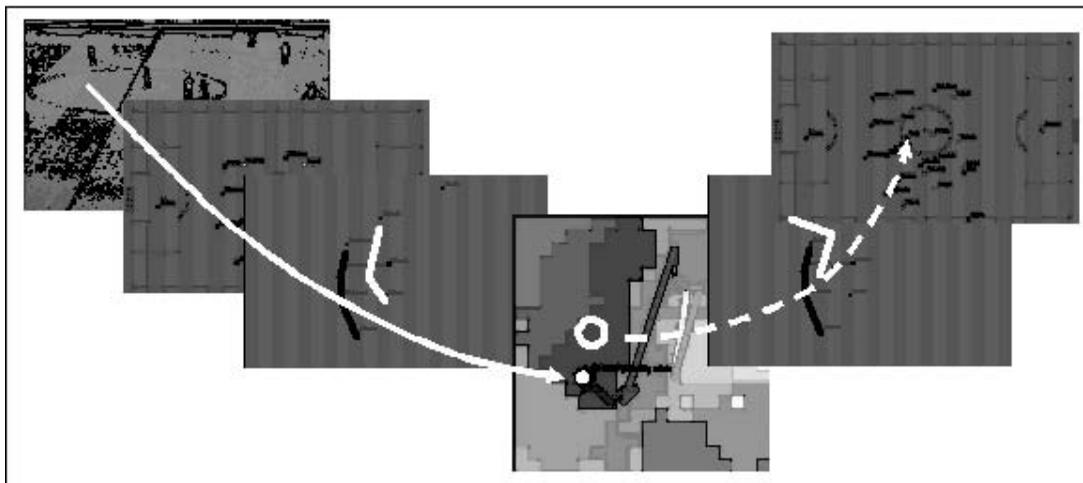
The result of about 30 years of work in this area is that a lot of problems could be solved methodically. The bottleneck, however, was the recording of data and the transfer to information. Meanwhile, data from biomechanical, physiological, and medical applications can be recorded automatically, and even in games like soccer automatic position recording has become possible. Therefore the problem has changed from "how to get data" to "how to transfer data to information".

The presented net-based approaches show how this problem can be handled, opening new perspectives of transferring theoretical approaches to practical work.

REFERENCES

Fritzke, B. (1997). A self-organizing network that can follow non-stationary distributions. *Proceedings of ICANN97, International Conference on Artificial Neural Networks*. 613-618.

Jäger, J., Perl, J. & Schöllhorn, W. (2007). Analysis of players' configurations by means of artificial neural networks. *International Journal of Performance Analysis of Sport*, 3 (7), 90-103.

Kohonen T. (1995). *Self-Organizing Maps*. Berlin–Heidelberg–New-York: Springer.

Lees, A., Barton, B. & Kerschaw, L. (2003). The use of Kohonen neural network analysis to establish characteristics of technique in soccer kicking. *Journal of Sports Sciences*, 21, 243-244.

Leser, R. (2006). Prozessanalyse im Fußball mittels Neuronaler Netze. M. Raab, A. Arnold, K. Gärtner, J. Köppen, C. Lempertz, N. Tielemann, H. Zastrow (Eds.), *Human Performance and Sport,* 2, 199-202.

McGarry, T., & Perl, J. (2004). Models of sports contests – Markov processes, dynamical systems and neural networks. M. Hughes, & I. M. Franks (Eds.), *Notational Analysis of Sport*, 227–242.

Perl, J. & Baca, A. (2003). Application of neural networks to analyze performance in sports. In E. Müller, H. Schwameder, G. Zallinger & V. Fastenbauer (Eds.), *Proceedings of the 8th annual congress of the European College of Sport Science*, 342.

Perl, J. & Dauscher, P. (2006). Dynamic Pattern Recognition in Sport by Means of Artificial Neural Networks. R. Begg & M. Palaniswami (Eds.), *Computational Intelligence for Movement Science*, 299-318.

Perl, J. & Lames, M. (2000). Identifikation von Ballwechselverlaufstypen mit Neuronalen Netzten am Beispiel Volleyball. W. Schmidt & A. Knollenberg (Eds.), *Schriften der dvs*, 112, 211-215.

Perl, J. & Weber, K. (2004). A Neural Network approach to pattern learning in sport. *International Journal of Computer Science in Sport*, 3 (1), 67-70.

Perl, J, Memmert, D., Bischof, J. & Gerharz, Ch. (2006). On a First Attempt to Modelling Creativity Learning by Means of Artificial Neural Networks. *International Journal of Computer Science in Sport*, 5 (2), 33-37.

Perl, J. (2002 a). Adaptation, Antagonism, and System Dynamics. G. Ghent, D. Kluka & D. Jones (Eds.), *Perspectives – The Multidisciplinary Series of Physical Education and Sport Science*, 4, 105-125.

Perl, J. (2002 b). Game analysis and control by means of continuously learning networks. *International Journal of Performance Analysis of Sport*, 2, 21-35.

Perl, J. (2004). A Neural Network approach to movement pattern analysis. *Human Movement Science*, 23, 605-620.

Pfeiffer, M. & Perl, J. (2006). Analysis of Tactical Structures in Team Handball by Means of Artificial Neural Networks. International *Journal of Computer Science in Sport*, 5 (1), 4-14.

Schöllhorn, W. (2004). Applications of artificial neural nets in clinical biomechanics. *Clinical Biomechanics*, 19 (9), 876-898.

KEY TERMS

Cluster: A collection of neurons is called a cluster, if they are similar and locally neighboured. Due to the topology preserving property of KfM-training classes of similar training vectors are mapped to clusters of neighboured neurons.

DyCoN: A DyCoN is a KFM-type network, where each neuron contains an individual PerPot-based self-control of its activation radius and learning rate. The DyCoN-concept enables for continuous learning and therefore supports continuous training and testing, training in phases and with generated data, on line-adaptation during tests and analyses, and flexible adaptation to new information patterns (Perl, 2002 a). (Note that DyCoN is used commercially. Therefore, technical details cannot be published but are under secrecy by DyCoS GmbH (www.dycos.net)).

DyCoNG: The concept of DyCoNG combines the concepts of DyCoN and GNG and completes it by dynamically generating "quality" neurons in order to represent relevant and rare information during the training process (Perl et al., 2006).

GNG: A GNG is network without a fixed neuron topology, which is able to generate new neurons on demand. Therefore a GNG is able to dynamically adapt its neuron structure to amount and structure of the trained information (Fritzke, 1997).

Information Pattern: An information pattern is a structure of information units like e.g. a vector or matrix of numbers, a stream of video frames, or a distribution of probabilities.

KFM: A KFM consists of a (normally: 2-dimensional) matrix of neurons, each of which contains a vector of attributes. Two neurons are called similar if the (Euclidian) distance of their attribute vectors is below a given threshold. Two neurons are called neighboured if they are next to each other regarding the given net topology (see Kohonen, 1995).

PerPot: PerPot is a model of dynamic adaptation, where an input flow feeds an internal strain potential as well as an internal response potentials, from which an output potential is fed by specifically delayed flows. Since the strain flow is negative and the response flow is positive, resulting in an oscillating stabilizing adaptation, the model is called antagonistic (Perl, 2002 a).

Test: In a test, an attribute vector is fed to the network to determine its type – i.e. the neuron it is corresponding to.

Training: During the training, attribute vectors are fed to the network and mapped to the corresponding neuron the entry of which is most similar to that of the attribute vector. After the training, the space of training attribute vectors is (more or less) completely represented by the neurons of the network – meaning that every training attribute vector belongs to a neuron the entry of which it is most similar to.

Type: The collection of attribute vectors that, after training, is represented by a neuron is called its type. Also the representing neuron can be called the type.

Neural Networks and Equilibria, Synchronization, and Time Lags

Daniela Danciu
University of Craiova, Romania

Vladimir Răsvan
University of Craiova, Romania

INTRODUCTION

All neural networks, both natural and artificial, are characterized by two kinds of dynamics. The first one is concerned with what we would call "learning dynamics", in fact the sequential (discrete time) dynamics of the choice of synaptic weights. The second one is the intrinsic dynamics of the neural network viewed as a *dynamical system* after the weights have been established *via* learning. Regarding the second dynamics, the emergent computational capabilities of a *recurrent neural network* can be achieved provided it has *many equilibria*. The network task is achieved provided it approaches these equilibria. But the *dynamical system* has a dynamics induced *a posteriori* by the learning process that had established the synaptic weights. It is not compulsory that this *a posteriori* dynamics should have the required properties, hence they have to be checked separately.

The standard stability properties (Lyapunov, asymptotic and exponential stability) are defined for a single equilibrium. Their counterpart for *several equilibria* are: *mutability*, *global asymptotics*, *gradient behavior*. For the definitions of these general concepts the reader is sent to Gelig *et. al.*, (1978), Leonov *et. al.*, (1992).

In the last decades, the number of *recurrent neural networks*' applications increased, they being designed for classification, identification and complex image, visual and spatio-temporal processing in fields as engineering, chemistry, biology and medicine (see, for instance: Fortuna *et. al.*, 2001; Fink, 2004; Atencia *et. al.*, 2004; Iwahori *et. al.*, 2005; Maurer *et. al.*, 2005; Guirguis & Ghoneimy, 2007). All these applications are mainly based on the existence of *several equilibria* for such networks, requiring them the "good behavior" properties above discussed.

Another aspect of the qualitative analysis is the so-called *synchronization* problem, when an external stimulus, in most cases periodic or almost periodic has to be tracked (Gelig, 1982; Danciu, 2002). This problem is, from the mathematical point of view, nothing more but existence, uniqueness and global stability of forced oscillations.

In the last decades the neural networks dynamics models have been modified once more by introducing the transmission **delays**. The standard model of a Hopfield-type network with delay as considered in (Gopalsamy & He, 1994) is

$$\frac{du_i}{dt} = -a_i u_i(t) + \sum_{j=1}^{n} w_{ij} g_j(u_j(t - \tau_{ij})) + I_i \quad i = \overline{1, n} \quad (1)$$

The present paper aims to a general presentation, with both research and educational purposes, of the three topics mentioned previously.

BACKGROUND

Dynamical systems with *several equilibria* occur in such fields of science and technology as electrical machines, chemical reactions, economics, biology and, last but not least, neural networks.

For systems with *several equilibria* the usual local concepts of stability are not sufficient for an adequate description. The so-called "global phase portrait" may contain both stable and unstable equilibria: each of them may be characterized separately since stability is a local concept dealing with a specific trajectory. But global concepts are also required for a better system description and this is particularly true for the case of the neural networks. Indeed, the neural networks may be viewed as interconnections of simple computing elements whose computational capability is increased by interconnection ("emergent collective capacities"

– to cite Hopfield). This is due to the nonlinear characteristics leading to the existence of several stable equilibria. The network achieves its computing goal if no self-sustained oscillations are present and it always achieves some steady-state (equilibrium) among a finite (while large) number of such states.

This behavior is most suitably described by the concepts arising from the papers of Kalman (1957) and Moser (1967). The last of them relies on the following remark concerning the rather general nonlinear autonomous system

$$\dot{x} = -f(x), \quad x \in \mathbb{R}^n \tag{2}$$

where $f(x) = \text{grad } G(x)$ and $G : \mathbb{R}^n \to \mathbb{R}$ is such that the number of its critical points is finite and is radially unbounded i.e. $\lim_{|x|\to\infty} G(x) = \infty$. Under these assumptions any solution of (2) approaches asymtotically one of the equilibria (which is also a critical point of G – where its gradient, i.e. f vanishes). Obviously the best limit behavior of a neural network would be like this – naturally called **gradient like behavior**. Nevertheless there are other properties that are also important while weaker; in the following we shall discuss some of them.

The mathematical object will be in the following the system of ordinary differential equations

$$\dot{x} = f(x,t) \tag{3}$$

and we shall first define some basic notions.

Definition 1 *a) Any constant solution of (3) is called* **equilibrium**; *the set of equilibria* E *is called* **stationary set**. *b) A solution of (3) is called* **convergent** *if it approaches asymptotically some equilibrium:*

$$\lim_{t\to\infty} x(t) = c \in E \ . \tag{4}$$

A solution is called **quasi-convergent** *if it approaches asymptotically the stationary set:*

$$\lim_{t\to\infty} d(x(t), E) = 0 , \tag{5}$$

*with d(z, **M**) being the distance (in the usual sense) from the point z to the set **M**.*
c) System (3) is called **monostable (strictly mutable)** *if every bounded solution is convergent (in the above*

sense); it is called **quasi-monostable** *if every bounded solution is quasi-convergent.*
d) System (3) is called **gradient-like** *if every solution is convergent; it is called* **quasi-gradient-like** *(has* **global asymptotics***) if every solution is quasi-convergent.*

Remark that *convergence* is a solution property while *monostability* and *gradient* property are associated to systems. For autonomous (time invariant) systems of the form (2) the following Lyapunov type results are available.

Lemma 1 *Consider system (2) and assume existence of a continuous function $V : \mathbb{R}^n \to \mathbb{R}$ that is nonincreasing along any of its solutions. If, additionally, a bounded on $\mathbb{R}^+$ solution x(t) for which there exists some $\tau > 0$ such that $V(x(\tau)) = V(x(0))$ is an equilibrium, then the system is quasi-monostable.*

Lemma 2 *If the assumptions of Lemma 1 hold and, additionally, V(x) is radially unbounded then system (2) is quasi-gradient like.*

Lemma 3 *If the assumptions of Lemma 2 hold and the set* E *is discrete (i.e. it consists of isolated points only) then system (2) is gradient-like.*

DYNAMICS ISSUES OF RECURRENT NEURAL NETWORKS

Neural Networks as Systems with Several Equilibria

It has been already mentioned that the emergent computational capacities of the neural networks are ensured by: a) nonlinear behavior of the neural cells; b) their connectivity. These two properties define the neural networks as **dynamical systems with many equilibria** whose performance depends on the (high) number of these equilibria and on the **gradient like property** of the network.

On the other hand, the standard **recurrent neural networks** (Bidirectional Associative Memory (Kosko, 1988), Hopfield (1982), cellular (Chua & Yang, 1988), Cohen-Grossberg (1983)), which contain internal feedback loops - having thus the propensity for instability, possess some "natural", i.e. associated in a natural way,

Lyapunov function allowing to obtain the required qualitative properties (Răsvan, 1998).

One of the most general models of neural networks that has a natural Lyapunov function is the Cohen-Grossberg model described by

$$\dot{x}_i = a_i(x_i)\left[b_i(x_i) - \sum_{j=1}^{n} c_{ij} d_j(x_j)\right], \qquad i = \overline{1,n}, \quad (6)$$

with $c_{ij} = c_{ji}$; this model may be written as

$$\dot{x} = -A(x) grad V(x) \qquad (7)$$

where $A(x)$ is a diagonal matrix with the entries

$$A_{ij}(x) = \frac{a_i(x_i)}{d_i'(x_i)}\delta_{ij} \qquad (8)$$

and $V : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$V(x) = \frac{1}{2}\sum_{1}^{n}\sum_{1}^{n} c_{ij} d_i(x_i) d_j(x_j) - \sum_{1}^{n}\int_{0}^{x_i} b_i(\lambda) d_i'(\lambda) d\lambda \qquad (9)$$

The presence of $A(x)$ makes system (7) a *pseudo-gradient system* – compare to (2).

The properties of the associated *Lyapunov function* (9) will give sufficient conditions in order to obtain the required qualitative behaviors for the system. The derivative function of (9) is:

$$W(x) = -\sum_{1}^{n} a_i(x_i) d_i'(x_i)\left[b_i(x_i) - \sum_{j=1}^{n} c_{ij} d_j(x_j)\right]^2 \le 0 \qquad (10)$$

One can see that the inequality (10) holds provided $a_i(\lambda) > 0$ and $d_i(\lambda)$ are monotone nondecreasing. If additionally $d_i(\lambda)$ are strictly increasing, then the set where $W = 0$ consists of equilibria only. The system results *quasi-gradient like* i.e. every solution approaches asymptotically the stationary set.

Consider now a model of artificial neural network implemented by electrical circuits:

$$R_i C_i \frac{dv_i}{dt} = -v_i + \sum_{j=1}^{n} \frac{R_i}{R_{ij}}\left(\varphi_j(v_j) - v_j\right) + R_i I_i \qquad (11)$$

with $\varphi_j(\cdot)$ being sigmoidal. Since sigmoidal functions are subject to sector restrictions and global Lipschitz inequalities, it was only natural to try to improve the stability conditions using the *Lyapunov functions* suggested by the Popov frequency domain inequalities and the Yakubovich-Kalman-Popov lemma. For instance, in (Danciu & Răsvan, 2000) there was considered a rather general system with several sector restricted nonlinearities and the Lyapunov function was constructed in a rational way starting from an improved frequency domain stability inequality of Popov type with PI multiplier.

In the case of (11) this rather involved approach gives a *gradient like behavior* provided the symmetry condition $R_{ij} = R_{ji}$ is observed.

Time Delays in Neural Networks

We shall consider here the model (1). Since we do not dispose (yet) in the time delay case of an instrument like the Lyapunov like lemmas given in BACKGROUND, we have to restrict ourselves to the analysis of the stability of a particular equilibrium.

If $\bar{u}_i, i = 1,\ldots,n$ is some equilibrium of (1) and if the deviations $z_i = u_i - \bar{u}_i$ are considered, the system in deviations is obtained

$$\frac{dz_i}{dt} = -a_i z_i(t) - \sum_{j=1}^{n} w_{ij}\varphi_j(z_j(t - \tau_{ij})), \quad i = \overline{1,n} \qquad (12)$$

with $\varphi_j(z_j) = g_j(\bar{u}_j) - g_j(\bar{u}_j + z_j)$. As known, if $g_j : \mathbb{R} \mapsto \mathbb{R}$ satisfy the usual sigmoid conditions i.e. $g_j(0) = 0$, monotonically increasing and globally Lipschitz - that is

$$0 \le \frac{g_j(\sigma_1) - g_j(\sigma_2)}{\sigma_1 - \sigma_2} \le L_j, \quad \forall \sigma_1 \ne \sigma_2, \qquad (13)$$

then φ_j defined above are such. With the usual notations of the field, let $z_t(\cdot) = z(t + \cdot)$ denote the state of (12) at t; the state space will be considered $\mathcal{C}(-r,0; \mathbb{R}^n)$ with $r = \max_{i,j}\tau_{ij}$, the space of continuous $\mathbb{R}^n$- valued mappings defined on $[-r, 0]$ with the usual norm of the uniform convergence. One considers the Lyapunov-Krasovskii functional (the analogue of the *Lyapunov function* of the delayless case) suggested by (Nishimura & Kitamura, 1969), $V : \mathcal{C} \mapsto \mathbb{R}_+$ as

N

$$W(u) = \sum_{i=1}^{n}\left[-a_i \pi_i u_i^2(0) - \pi_i u_i(0) \sum_{j=1}^{n} w_{ij} f_j\left(u_j(-\tau_{ij}) \right) \right]$$

$$+ \sum_{1}^{n}\sum_{1}^{n} \rho_{ij}\left[u_j^2(0) - u_j^2(-\tau_{ij}) \right]$$

$$(19)$$

is negative definite with a quadratic upper bound. Then the system (17) has a unique global solution $\overline{u}_i(t)$, $i = \overline{1,n}$ *which is bounded on* $\mathbb{R}$ *and exponentially stable. Moreover, this solution is periodic or almost periodic according to the character of* $c_i(t)$*- periodic or almost periodic respectively.*

FUTURE TRENDS

Supposing the field of AI has its own dynamics, the neural networks and their structures will evolve in order to improve the imitative behavior i.e. more of the "natural" intelligence will be transferred to AI. Consequently, science and technology will deal with new structures of various physical natures having ***multiple equilibria***. At least the following qualitative behaviors will remain under study: stability-like properties (dichotomy, ***gradient behavior*** a.s.o.), ***synchronization*** (forced oscillations, almost linear behavior, chaos control) and complex dynamics (including chaotic behavior).

CONCLUSIONS

Our experience on neural networks dynamics shows that the most important study is to obtain conditions for gradient or quasi-gradient like behavior. Besides the comparison method of (Popov, 1979) which requires relaxation of the condition of the identical dynamics of all neurons, the most popular tool remains the Lyapunov method.

If the Lyapunov like lemmas given in BACK-GROUND would be available in the time delay case, then improved Lyapunov functionals remaining constant on the set of equilibria could ensure a gradient like behavior.

REFERENCES

Atencia, M., Joya, G., Sandoval, F. (2004). Parametric identification of robotic systems with stable time-varying Hopfield networks. *Neural Computing and Applications*, Springer London, 13(4), 270-280.

Chua, L. & Yang, L. (1988). Cellular neural networks: theory and applications, *IEEE Transactions on Circuits and Systems*, CAS-35, 1257-1290.

Cohen, M. A. & Grossberg, S. (1983). Absolute stability of pattern formation and parallel storage by competitive neural networks. *IEEE Transactions of Systems, Man & Cybernetics,* 13, 815-826.

Danciu, D. (2002). Qualitative behavior of the time delay Hopfield type neural networks with time varying stimulus. *Annals of The University of Craiova*, Series: Electrical Engineering 26, 72–82.

Danciu, D. & Răsvan, V. (2000). On Popov-type stability criteria for neural networks. *Electronic Journal on Qualitative Theory of Differential Equations* 23. http://www.math.uszeged.hu/ejqtde/6/623.pdf

Danciu, D. & Răsvan, V. (2007). Dynamics of Neural Networks – Some Qualitative Properties. *Computational and ambient Intelligence.* Lectures Notes in Computer Science, (4507), F. Sandoval, A. Prieto, J. Cabestany, editors, 8-15.

Fink, W. (2004). Neural attractor network for application in visual field data classification. *Physics in Medicine and Biology*, 49(13)**,** 2799-2809.

Fortuna, L., Arena, P., Balya, D. & Zarandy, A. (2001). Cellular Neural Networks. *IEEE Circuits and Systems Magazine*, 4, 6–21.

Gelig, A. Kh., Leonov, G. A. & Yakubovich, V.A. (1978). *Stability of nonlinear systems with non-unique equilibrium state*. (in Russian) U.R.S.S.: Moscow, Nauka Publishers House.

Gelig, A. Kh. (1982) *Dynamics of pulse systems and neural networks* (in Russian). Leningrad Univ. Publishing House.

Gopalsamy, K. & He, X. Z. (1994). Stability in asymmetric Hopfield nets with transmission delays, *Physica D.*, 76, 344-358.

Guirguis, L.A., Ghoneimy, M.M.R.E. (2007). Channel Assignment for Cellular Networks Based on a Local Modified Hopfield Neural Network. *Wireless Personal Communications*, Springer US, 41(4), 539-550.

Halanay, A. (1967). Invariant manifolds for systems with time lag. *Differential and dynamical systems*. Hale & La Salle editors, New York, Academic Press, 199–213.

Hale J. K. & Verduyn Lunel, S. M. (1993). *Introduction to Functional Differential Equations*. Springer-Verlag.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academic Science U.S.A.*, 79, 2554-2558.

Iwahori, Y., Kawanaka, H., Fukui, S., Funahashi, K. (2005). Obtaining Shape from Scanning Electron Microscope using Hopfield Neural Network. *Journal of Intelligent Manufacturing*, Springer US, 16(6), 715-725.

Kalman, R. E. (1957). Physical and mathematical mechanisms of instability in nonlinear automatic control systems. *Transactions American Society of Mechanical Engineers*, 79(3).

König, P. & Schillen, J.B. (1991). Stimulus dependent assembly formation of oscillatory responses: I. Synchronization. *Neural Computation*, 3, 155-166.

Kopell, N. (2000). We got rhythm: dynamical systems of the nervous system. *Notices of American Mathematical Society*, 47, 6-16.

Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions Systems, Man and Cybernetics,* 18, 49-60.

Leonov, G.A., Reitmann, V. & Smirnova, V.B. (1992). *Non-local methods for pendulum-like feedback systems*, Germany: Leipzig, Teubner Verlag.

Maurer, A., Hersch, M. and Billard, A. G. (2005). Extended Hopfield Network for Sequence Learning: Application to Gesture Recognition. *Proceedings of 15th International Conference on Artificial Neural Network*, 493-498.

Moser, J. (1967). On nonoscillating networks. *Quarterly Applied Mathematics,* 25, 1-9.

Nishimura, M., Kitamura, S. & Hirai, K. (1969). A Lyapunov Functional for Systems with Multiple Non-linearities and Time Lags, Technological Reports, Japan: Osaka University, 19(860), 83-88.

Popov, V.M. (1979). Monotonicity and Mutability. *Journal of Differential Equations*, *31*(3), 337-358.

Răsvan, V. (1998). Dynamical Systems with Several Equilibria and Natural Lyapunov Functions, *Archivum mathematicum*, 34(1), [EQUADIFF 9], 207-215.

KEY TERMS

Asymptotic Stability: The solution $\bar{x}(t)$ of (3) is called *asymptotically stable* if it is Lyapunov stable (see below) and, moreover, there exists $\delta_0 > 0$ such that if $|x_0 - \bar{x}(t_0)| < \delta_0$ then $\lim_{t\to\infty}|x(t;t_0,x_0) - \bar{x}(t)| = 0$.

Fixed Point Theorem: If $f(x)$ is some function of real variable with real values, the values such that $f(x) = x$ are called the *fixed points* of the mapping. In general, if $f : \mathbf{X} \mapsto \mathbf{X}$ is a mapping from the metric space $\mathbf{X}$ inf : X X into itself, the fixed points of this mapping are defined as above. A *fixed point theorem* is a theorem showing under which conditions some mapping has a fixed point in the corresponding metric space.

Frequency Domain Stability Inequality of Popov: Consider a feedback structure containing a linear dynamical block with the transfer function $H(s)$ and a nonlinear function - subject to the sector condition $0 < \phi(\sigma)\sigma < k\sigma^2$. *The Popov inequality* ensures absolute stability i.e. global asymptotic stability of the zero equilibrium for all nonlinear functions satisfying the above inequality and reads as follows: there exists some β such that

$$\frac{1}{k} + \Re e\left(1 + j\omega\beta\right)H(j\omega) > 0, \quad \forall \omega \in \mathbb{R}$$

.

Global Stability: An equilibrium is *global (asymptotically) stable* if it is the unique equilibrium of the dynamical system and the property holds globally (its domain of attraction is the entire state space).

Lyapunov Function: State scalar function defined on the state space of a system in order to obtain some qualitative properties - stability of equilibria, oscillatory

behavior etc. - using a single function instead of several i.e. system's state trajectories. A Lyapunov function is usually positive definite and, along system's trajectories, is at least nonincreasing. The definite sign condition may also be relaxed for the generalized Lyapunov functions in the LaSalle sense. The basic physical model for the Lyapunov function is system's energy - a state function that is nonincreasing along the state trajectory being at the same time positive definite. The strength of the Lyapunov function is exactly its independence of the physical concepts since writing down the stored energy of a system is not an easy job except possibly such standard cases as mechanical systems or electrical circuits. The energy like concepts may be nevertheless inspiring when "guessing" a Lyapunov function. In the infinite dimensional cases e.g. time delay or propagation systems, the Lyapunov function is replaced by a *Lyapunov functional* defined on the infinite dimensional state space.

Oscillations (Self-Sustained and Forced): Type of steady state behavior when the state trajectories, while remaining bounded, never reach an equilibrium but their deviations from this equilibrium keep sign changing. Usually an oscillation is viewed as having some recurrent properties, being either periodic or almost periodic. When the system is autonomous i.e. free of external oscillatory signals while nevertheless displaying an oscillatory behavior which is sustained by non-oscillatory internal factors of the system, it is said that this system displays *self-sustained oscillations* (the term belongs to Mandelstamm and Andronov). When the system is non-autonomous and subject to external oscillatory signals (*stimuli*), the limit regime that occurs is called *forced oscillation*.

Phase Portrait: Term borrowed from the Poincaré theory of the phase (space) plane where this portrait is better defined. Its extension to higher order systems is mainly informal, based on geometric arguments. By *phase portrait* it is understood the total of state trajectories as limit regimes (equilibria, recurrent motions, limit sets) and standard trajectories e.g. defined by initial conditions.

Recurrent Neural Network (RNN): Neural networks which display feedback interconnections among their units (neurons). Due to these cyclic connections RNNs are nonlinear dynamical systems with very rich spatial and temporal behaviors: stable and unstable fixed points, limit cycles and chaotic behavior. These behaviors make them suitable for modeling certain cognitive functions such as associative memory, unsupervised learning, self-organizing maps and temporal reasoning.

Synchronization: Interaction phenomenon among coupled subsystems of a system resulting in some ordering of their evolution. Its maximal stage is the complete synchronization of the subsystems' periods resulting in a periodic evolution of the state of the entire system. When a system is externally forced by an oscillatory signal, synchronization means a limit regime of the entire state, which has the same waveform as the forcing signal (periodic with the same period if the forcing signal is periodic or almost periodic if the forcing signal is such).

Stability: Qualitative property of the solution of a system with the significance of the limitation of the perturbations effect on the considered solution viewed as basic. Among all kinds of stability (bounded input/bounded output, Lagrange stability, Birkhoff stability, input-to-state stability) the stability in the sense of Lyapunov - with respect to the initial conditions, viewed as incorporating the effect of short-period perturbations - is the most widely used; it means that sufficiently small deviations in the initial condition (state) will result in arbitrarily small deviations in the current state at all following moments. Rigorously, the basic solution $\bar{x}(t)$ of (3) is called *stable in the sense of Lyapunov* if, for any $\varepsilon > 0$ arbitrarily small and any $t_0 \in \mathbb{R}$ there exists some $\delta(\varepsilon, t_0) > 0$ sufficiently small such that if $|x_0 - \bar{x}(t_0)| < \delta(\varepsilon, t_0)$, then $|x(t; t_0, x_0) - \bar{x}(t)| < \varepsilon$ for all $t > t_0$. If in the above definition δ is independent of the initial moment t_0 the stability is called uniform; from the point of view of the practice, this is the more important stability notion of stability. It is also a necessary condition for uniform asymptotic stability (see above).

Neural Networks and HOS for Power Quality Evaluation

Juan J. González De la Rosa
Universities of Cádiz-Córdoba, Spain

Carlos G. Puntonet
University of Granada, Spain

A. Moreno-Muñoz
Universities of Cádiz-Córdoba, Spain

INTRODUCTION

Power quality (PQ) *event* detection and classification is gaining importance due to worldwide use of delicate electronic devices. Things like lightning, large switching loads, non-linear load stresses, inadequate or incorrect wiring and grounding or accidents involving electric lines, can create problems to sensitive equipment, if it is designed to operate within narrow voltage limits, or if it does not incorporate the capability of filtering fluctuations in the electrical supply (Gerek et. al., 2006; Moreno et. al., 2006).

The solution for a PQ problem implies the acquisition and *monitoring* of long data records from the energy distribution system, along with an automated detection and classification strategy which allows identify the cause of these voltage anomalies. Signal processing tools have been widely used for this purpose, and are mainly based in spectral analysis and wavelet transforms. These *second-order* methods, the most familiar to the scientific community, are based on the independence of the spectral components and evolution of the spectrum in the time domain. Other tools are threshold-based algorithms, linear classifiers and Bayesian networks. The goal of the signal processing analysis is to get a feature vector from the data record under study, which constitute the input to the computational intelligence modulus, which has the task of classification. Some recent works bring a different strategy, based in higher-order *statistics* (HOS), in dealing with the analysis of *transients* within PQ analysis (Gerek et. al., 2006; Moreno et. al., 2006) and other fields of Science (De la Rosa *et. al*., 2004, 2005, 2007).

Without perturbation, the 50-Hz of the voltage *waveform* exhibits a *Gaussian* behaviour. Deviations from Gaussianity can be detected and characterized via HOS. Non-Gaussian processes need third and fourth order statistical characterization in order to be recognized. In order words, *second-order* moments and *cumulants* could be not capable of differentiate non-*Gaussian events*. The situation described matches the problem of differentiating between a transient of long duration named fault (within a signal period), and a short duration transient (25 per cent of a cycle). This one could also bring the 50-Hz voltage to zero instantly and, generally affects the sinusoid dramatically. By the contrary, the long-duration transient could be considered as a modulating signal (the 50-Hz signal is the carrier). These *transients* are intrinsically non-stationary, so it is necessary a battery of observations (sample registers) to obtain a reliable characterization.

The main contribution of this work consists of the application of higher-order central cumulants to characterize PQ *events*, along with the use of a competitive layer as the classification tool. Results reveal that two different clusters, associated to both types of *transients*, can be recognized in the 2D graph. The successful results convey the idea that the physical underlying processes associated to the analyzed *transients*, generate different types of deviations from the typical effects that the noise cause in the 50-Hz sinusoid voltage *waveform*.

The paper is organized as follows: Section on higher-order cumulants summarizes the main equations of the cumulants used in the paper. Then, we recall the competitive layer's foundations, along with the *Kohonen* learning rule. The experience is described then, and the conclusions are drawn.

HIGHER-ORDER CUMULANTS

High-order *statistics*, known as *cumulants*, are used to infer new properties about the data of *non-Gaussian* processes (Mendel, 1991; Nikias & Mendel, 2003). The relationship among the cumulants of *r* stochastic signals, $\{x_i\}_{i\in[1,r]}$, and their moments of order *p, p ≤ r*, can be calculated by using the *Leonov-Shiryayev* formula (Nandi, 1999; Nikias & Mendel, 2003). For an *r*th-order stationary random process *{x(t)}*, the *r*th-order cumulant is defined as the joint *r*th-order cumulant of the random variables $x(t)$, $x(t+\tau_1)$, …, $x(t+\tau_{r-1})$,

$$C_{r,x}(\tau_1, \tau_2, ..., \tau_r) = Cum[x(t)\, x(t+\tau_1)...x(t+\tau_r)]. \qquad (1)$$

Considering $\tau_1 = \tau_2 = \tau_3 = 0$ in Eq. (1), we have some particular cases:

$$?_{2,x} = E\{x^2(t)\} = C_{2,x}(0), \qquad (2a)$$

$$?_{3,x} = E\{x^3(t)\} = C_{3,x}(0,0), \qquad (2b)$$

$$?_{4,x} = E\{x^4(t)\} - 3(?_{2,x})^2 = C_{4,x}(0,0,0). \qquad (2c)$$

Eqs. (2) are measurements of the variance, skewness and kurtosis of the statistical distribution, in terms of the *cumulants* at zero lags. We will use and refer to normalized quantities because they are shift and scale invariant.

COMPETITIVE LAYERS

The neurons in a competitive layer distribute themselves to recognize frequently presented input vectors. The competitive transfer function accepts a net input vector *p* for a layer (each neuron competes to respond to *p*) and returns outputs of 0 for all neurons except for the winner, which is associated with the most positive element of the net input. For zero bias, the neuron whose weight vector is closest to the input vector has the least negative net input and, therefore, wins the competition to output a 1.

The winning neuron will move closer to the input, after this has been presented. The weights of the winning neuron are adjusted with the *Kohonen* learning rule. If for example the *i*th-neuron wins, the elements of the *i*th-row of the input weight matrix (**IW**) are adjusted as shown in Eq. (3):

$$\mathbf{IW}_i^{1,1}(q) = \mathbf{IW}_i^{1,1}(q-1) + ?\, [p(q) - \mathbf{IW}_i^{1,1}(q-1)], \qquad (3)$$

where *p* is the input vector, *q* is the time instant, and α is the learning rate. The *Kohonen* rule allows the weights of a neuron to learn an input vector, so it is useful in recognition applications. The winning neuron is more likely to win the competition the next time a similar vector is presented. As more and more inputs are presented, each neuron in the layer closest to a group of input vectors soon adjusts its weight vector toward those inputs. Eventually, if there are enough neurons, every cluster of similar input vectors will have a neuron that outputs "1" when a vector in the cluster is presented.

EXPERIMENTAL RESULTS

The aim is to differentiate between two classes of PQ *events*, named long-duration and short-duration. The experiment comprises two stages. The feature extraction stage is based on the computation of cumulants. Each vector's coordinate corresponds to the local maximum and minimum of the 4th-order central *cumulant*. Secondly, the classification stage is based on the application of the competitive layer to the feature vectors. We use a two-neuron competitive layer, which receives two-dimensional input feature vectors during the network training.

We analyze a number of 16 1000-point real-life registers during the feature extraction stage. Before the computation of the cumulants, two pre-processing actions have been performed over the sample signals. First, they have been normalized because they exhibit very different-in-magnitude voltage levels. Secondly, a high-pass digital filter (5th-order Butterworth model with a characteristic frequency of 150 Hz) eliminates the low frequency components which are not the targets of the experiment. This by the way increases the non-*Gaussian* characteristics of the signals, which in fact are reflected in the higher-order cumulants. Fig. 1 shows the comparison of the two types of *events*.

After pre-processing, a battery of sliding central cumulants (2nd, 3rd and 4th-order) is calculated. Each cumulant is computed over 50 points; this window's length (50 points) has been selected neither to be so long to cover the whole signal nor to be very short. The algorithm calculates the 3 central cumulants over 50 points, and then it jumps to the following starting

Figure 1. Analysis for two types of transients
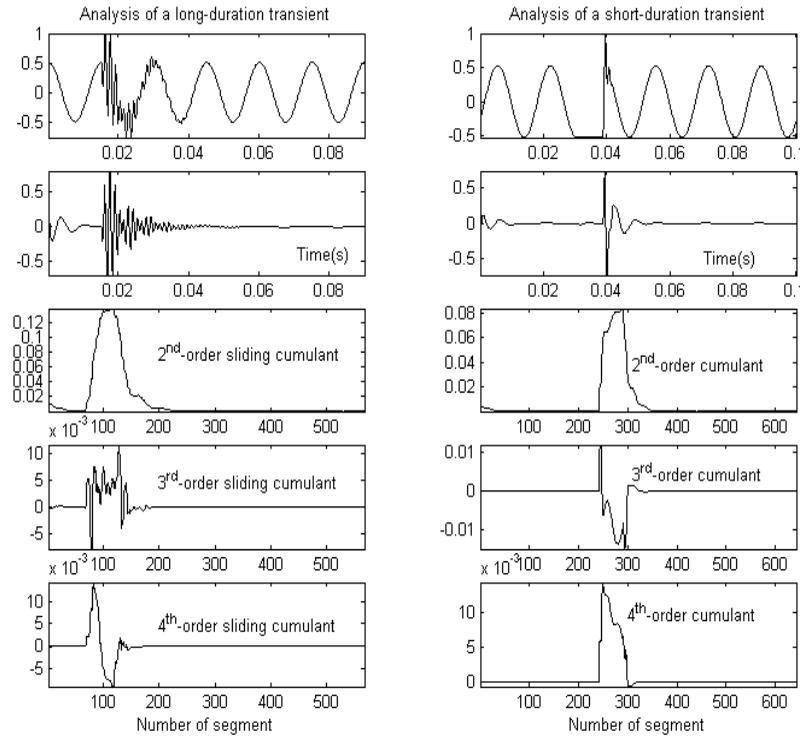
point; as a consequence we have 98 per cent overlapping sliding windows (49/50=0.98). Each computation over a window (called a segment) outputs 3 cumulants.

The signal processing analysis indicates that the 2nd-order cumulant sequence (the variance), clearly indicates the presence of an event. Both types of *transients* exhibit an increasing variance in the neighbourhood of the PQ event, that present the same shape, with only one maximum. The magnitude of this maximum is by the way the only available feature which can be used to distinguish different events from the second-order point of view.

Resulting from the classification stage, the bi-dimensional representation (2-dimensional feature vectors) suggests very intelligible 2-D graphs for 4th-order. 3rd-order diagrams don't show quite different clusters because maxima and minima are similar. It is possible to differentiate PQ *events* from the 3rd-order

perspective if we consider more features in the input vector (perhaps 3-D feature vectors), like the number of extremes (maxima and minima), and the order in which the maxima and the minima appear as time increases.

The sliding 4th-order cumulants exhibit clear differences, not only for the shape of the time-domain graphs, but also for the different location of minima, which suggest a clustering for the points in the 2-D feature space. Fig. 2 shows an example of 4th-order cumulant sequence comparison for the two types of *transients*. For each sample register (data record) the sliding 4th-order cumulants' sequence is calculated (as in Fig. 2). For each data record, the maximum and the minimum are detected and selected as a point in the feature space.

Fig. 3 presents the results of the training stage, using the *Kohonen* rule. The horizontal (vertical) axis cor-

Figure 2. Comparison of 4th-order cumulants' sequences for two types of transients
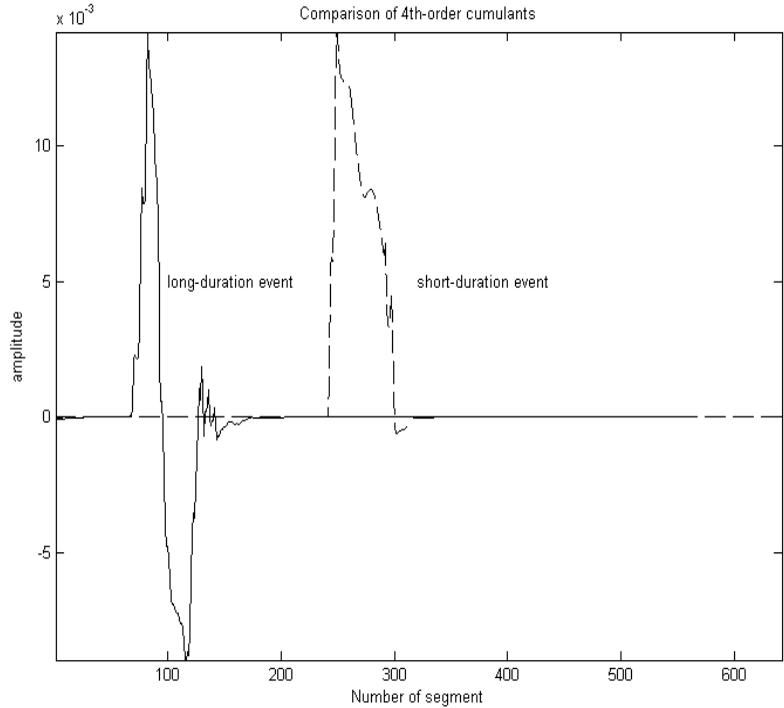
responds to the maxima (minima) values. Each cross in the diagram corresponds to an input vector and the circles indicate the final location of the weight vector (after learning) for the two neurons of the competitive layer. Before training, both weight vectors pointed to the asterisk, which is the initializing point (the midpoint of the input intervals).

The separation between classes (inter-class distance) is well defined. Both types of PQ events are clustered. The correct configuration of the clusters is corroborated during the simulation of the neural network, in which we have obtained an approximate classification accuracy of 97 percent. During the simulation, new signals (randomly selected from our data base) were processed using this methodology. The accuracy of the classification results increases with the number of data. To evaluate the confidence of the statistics a

significance test has been conducted. As a result, the number of measurements is significantly correct.

CONCLUSION

In this paper we have proposed an automatic method to detect and classify two PQ *transients*, named short and long-duration. The method comprises two stages. The first includes pre-processing (normalizing and filtering) and outputs the 2-D feature vectors, each of which coordinate corresponds to the maximum and minimum of the central cumulants. The second stage uses a neural network to classify the signals into two clusters. This stage is different-in-nature from the one used in (Gerek et. al., 2006) consisting of quadratic classifiers. The configuration of the clusters is assessed

Figure 3. Competitive layer training results over 20 epochs. Upper cluster: Short-duration PQ-events. Down cluster: Long-duration events.
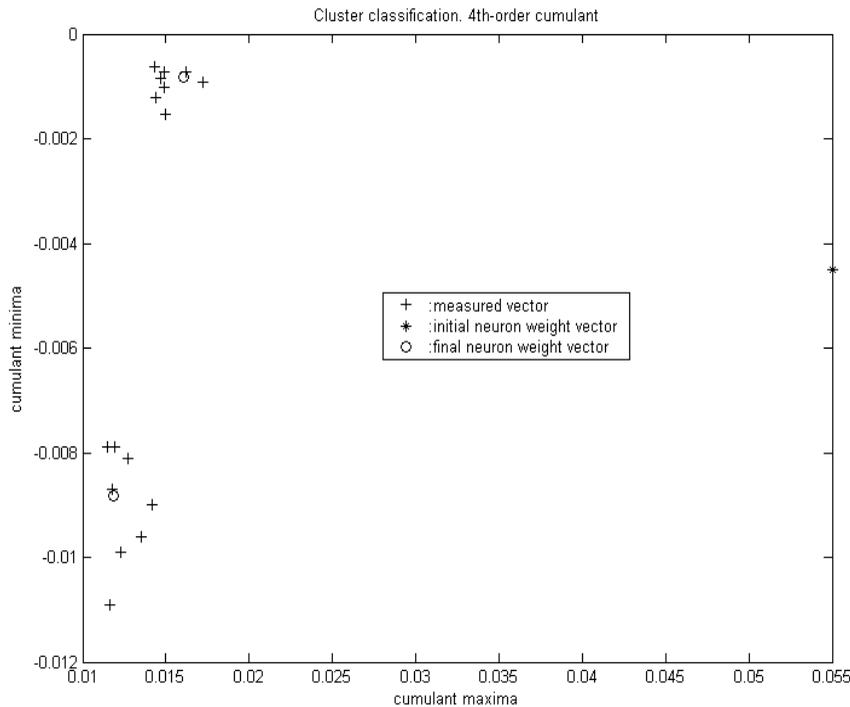
during the simulation of the network, in which we have obtained acceptable classification accuracy.

ACKNOWLEDGMENT

REFERENCES

Bendat, J., Piersol, A.: Random Data Analysis and Measurement Procedures, 3rd. Edition, Vol. 1 of Wiley Series in Probability and Statistics, Wiley Interscience, 2000.

Chonavel, T., *Statistical Signal Processing. Modelling and Estimation*, 1st ed., ser. Advanced Textbooks in Control and Signal Processing. London: Springer, 2003, vol. 1.

De la Rosa, J.J.G., Puntonet, C.G., Lloret, I., Górriz, J.M.: Wavelets and wavelet packets applied to termite detection. In: ICCS 2005. LNCS, vol. 3514, pp. 900–907. Springer, Heidelberg (2005)

De la Rosa, J.J.G., Ruzzante, J., Piotrkowski, R.: Third-order spectral characterization of acoustic emission signals in ring-type samples from steel pipes for the oil industry. In: Elsevier. (Ed.) Mechanical systems and Signal Processing, vol. 21, pp. 1917–1926 (2007)

De la Rosa, J.J.G., Lloret, I., Puntonet, C.G., Górriz, J.M.: Higher-order statistics to detect and characterise termite emissions. Electronics Letters 40, 1316–1317, Ultrasonics (2004)

De la Rosa, J.J.G., Puntonet, C.G., Lloret, I.: An application of the independent component analysis to monitor acoustic emission signals generated by termite activity in wood. In: Elsevier. (Ed.) Measurement, vol. 37, pp. 63–76 (2005)

De la Rosa, J.J.G, Moreno-Muñoz, A. Higher-order cumulants and spectral kurtosis for early detection of subterranean termites," Mechanical Systems and Signal Processing (Ed. Elsevier), vol. In Press, Accepted Manuscript, 2007, available online 1 September 2007.

Gerek, O.N., Ece, D.G.: Power-quality event analysis using higher order cumulants and quadratic classifiers. IEEE Transactions on Power Delivery 21, 883–889 (2006)

Mendel, J.M.: Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. In: Proceedings of the IEEE 79, 278–305 (1991)

Moreno, A., Pallarés, V., De la Rosa, J.J.G., Galisteo, P.: Study of voltage sag in a highly automated plant. In: MELECON 2006, Proceedings of the 2006 13th IEEE Mediterranean Electrotechnical Conference.

Moreno-Muñoz, A. and Mª D. Redel. Calm in the campus: power disturbances threaten university life. IEE Power Engineer, 19 (4), (2005), p. 34

Moreno-Muñoz, A.; Redel, M. D. and González, M. Power quality in high-tech campus. Proc. of the Institution of Mechanical Engineers, part A: Journal of Power and Energy. 220 (3), (2006) p. 257

Nandi, A.K.: Blind Estimation using Higher-Order Statistics, 1st Edn., vol. 1. Kluwer Academic Publichers, Boston (1999)

Nikias, C.L., Mendel, J.M.: Signal processing with higher-order spectra. IEEE Signal Processing Magazine, pp. 10–37 (1993)

Nikias, C.L., Petropulu, A.P.: Higher-Order Spectra Analysis. In: A Non-Linear Signal Processing Framework, Prentice-Hall, Englewood Cliffs, NJ (1993)

KEY TERMS

Artificial Neural Networks: A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Cluster: A set of incidences relative to the characteristics associated to some signals, which have been previously analyzed.

Cumulants: Statistics that characterize a probability distribution. A distribution with given cumulants can be approximated through the Edgeworth series.

Competitive Layer: The neurons in a competitive layer distribute themselves to recognize frequently presented input vectors.

HOS: Higher-Order Statistics; the set of statistics of order higher than 2. The advantage of using them is based on the advantage of noise rejection for symmetrically distributed processes.

Power Quality: Is the branch of research which aims to study the techniques for the assessment of the quality of electricity.

Transient: A signal which vanishes with the time and usually with short duration. They are very common in industry applications. *Transients* may occur either in repeatable fashion or as random impulses.

Neural Networks on Handwritten Signature Verification

J. Francisco Vargas
University of Las Palmas de Gran Canaria, Spain & Universidad de Antioquia, Colombia

Miguel A. Ferrer
University of Las Palmas de Gran Canaria, Spain

INTRODUCTION

Biometric offers potential for automatic personal identification and verification, differently from other means for personal verification; biometric means are not based on the possession of anything (as cards) or the knowledge of some information (as passwords). There is considerable interest in biometric authentication based on automatic signature verification (ASV) systems because ASV has demonstrated to be superior to many other biometric authentication techniques e.g. finger prints or retinal patterns, which are reliable but much more intrusive and expensive. An ASV system is a system capable of efficiently addressing the task of make a decision whether a signature is genuine or forger. Numerous pattern recognition methods have been applied to signature verification. Among the methods that have been proposed for pattern recognition on ASV, two broad categories can be identified: memory-based and parameter-based methods as a neural network. The Major approaches to ASV systems are the template matching approach, spectrum approach, spectrum analysis approach, neural networks approach, cognitive approach and fractal approach.

The proposed article reviews ASV techniques corresponding with approaches that have so far been proposed in the literature. An attempt is made to describe important techniques especially those involving ANNs and assess their performance based on published literature. The paper also discusses possible future areas for research using ASV.

BACKGROUND

As any human production, handwriting is subject to many variations from very diverse origins: Historic, geographic, ethnic, social, psychological, etc (Bou-letreau, 1998). ASV is a difficult problem because signature samples from the same person are similar but not identical. In addition, a person signature often changes radically during their lifetime (Hou, 2004). Although these factors can affect a given instance of a person writing, writing style develops as the writer learns to write, as do consistencies which are typically retained (Guo, 1997). One of the methods used by expert document examiners is to try to exploit these consistencies and identify ones which are both stable and difficult to imitate. In general, ASV systems can be categorized into two kinds: The On-line and Off-line systems. For On-line, the use of electronic devices to capture dynamics from signature permits to register more information about the signing process while improving the system performance, in the case of Off-line approaches for ASV, this dynamic information is lost and only a static image is available. This makes it quit difficult to define effective global or local features for the verification purpose.

Three different types of forgeries are usually take into account on ASV system: random forgeries, produced without knowing either the name of the signer nor the shape of his signature; simple forgeries, produced knowing the name of the signer but without having an example of his signature; and skilled forgeries, produced by people who, looking at an original instance of the signature, attempt to imitate it as closely as possible. The problem of signature verification become more difficult when passing from random to simple and skilled forgeries, the later being so difficult a task that even human beings make errors in several cases. It is pointing out that several systems proposed up to now, while performing reasonably well on a single category of forgeries, decrease in performance when working with all the categories simultaneously, and generally this decrement is bigger than one would expect.(Abuhaiba,2007; Ferrer,2005).

Numerous pattern recognition methods have been applied to signature verification (Plamondon, 1989). Among the methods that have been proposed for pattern recognition, two broad categories can be identified: memory-based techniques in which incoming patterns are matched to a (usually large) dictionary of templates, and parameter-based methods in which pre-processed patterns are sent to a trainable classifier such as a neural network (Lippmann, 1987). Memory-based recognition methods require a large memory space to store the templates, while a neural network is a parameter-based approach which just requires a small amount of memory space to store the linking weights among neurons. Mighell et al (Mighell, 1989) were apparently the first to work in applying NNs for off-line signature classification. Sabourin and Drouhard (Sabourin,1992) presented an method based on directional probability density functions together with a BackPropagation neural networks (BPN) to detect random forgery. Qi and Hunt (Qi, 1996) used global and grid features with a simple Euclidean distance classifier. Sansone and Vento (Sansone,2000) proposed a sequential three-stage multi-expert system, in which the first expert eliminates random and simple forgeries, the second isolates skilled forgeries, and the third gives the final decision by combining decisions of the previous stages together with reliability estimations. Baltzakis and Papamarkos (Baltzakis,2001) developed a two-stage neural network, in which the first stage gets the decisions from neural networks and Euclidean distance classifiers supplied by the global, grid and texture features, and the second combines the four decisions using a radial-base function (RBF) neural network.

MAIN FOCUS OF THE CHAPTER

As mentioned above, the major approaches to signature verification systems are the template matching approach, spectrum approach, spectrum analysis approach, neural networks approach, cognitive approach and fractal approach. The rigid template matching, the simplest and earliest approach to pattern recognition, can detect random forgeries from genuine signatures successfully, but cannot detect skilled forgeries effectively. The statistical approach, including HHMs, Bayesian and so on, can detect random forgeries as well as skilled forgeries from genuine ones. Structural approach shows good performance when detecting

genuine signatures and forgeries. But this approach may yield a combinatorial explosion of possibilities to be investigated, demanding large training sets and very large computational efforts. The spectrum analysis approach can be applied to different languages, including English and Chinese. Moreover it can be applied to either on-line or off-line verification systems.

Neural networks approach offers several advantages such as, unified approaches for feature extraction and classification and flexible procedures for finding good, moderately nonlinear solutions. When it is used in either on-line or off-line signature verification, it also shows reasonable performance.

Neural Networks on ASV

Multi-layer perceptron (MLP) neural networks are among the most commonly used classifiers for pattern recognition problems. Despite their advantages, they suffer from some very serious limitations that make their use, for some problems, impossible. The first limitation is the size of the neural network. It is very difficult, for very large neural networks, to get trained. As the amount of the training data increases, this difficulty becomes a serious obstacle for the training process. The second difficulty is that the geometry, the size of the network, the training method used and the training parameters depend substantially on the amount of the training data. Also, in order to specify the structure and the size of the neural network, it is necessary to know a priori the number of the classes that the neural network will have to deal with. Unfortunately, when talking about a useful ASV, a priori knowledge about the number of signatures and the number of the signature owners is not available (Baltzakis,2001).

For the BPN case, a learning law is used to modify weight values based on an output error signal propagated back through the network. From random initial values, the weights are changed according to this learning law that uses a learning rate and a smoothing rate which sometimes allows a faster convergence of the training phase. The training phase is critical, especially when the data to be classified are not clearly distinguishable and when there are not enough examples to conduct training. In this case, the training phase can be very long and it may even be impossible to obtain an acceptable performance. Usually a criterion for stopping the training phase is defined. After that, several rejection methods are evaluated to improve the decision taken by

this kind of classifier. Finally, the number of neurons in the hidden layer of the BPN is adjusted in order to increase the global performance of the first stage of the ASV (Drouhard, 1996).

An interesting aspect of BPN is that during learning process, the hidden layers build an internal representation of the inputs that is useful to produce the output (Looney, 1997). (Fleming. 1990) used a two-stage NN with the same number of neurones for input and output layers, and fewer units for the hidden layer. This forces the network to encode the inputs in a smaller dimensional space retaining most of the relevant information in an equivalent way as the Principal Component Analysis (PCA) method. This class of networks are known as compression networks. An important property of compression networks is that they can act as auto associative or content addressable memories (Kohonen, 1977; Valentin, 1994). This means that these networks are able to acceptably reconstruct a degraded pattern when noise is given as input or to complete an incomplete input pattern (O'Toole, 1993). The quality of the results will depend on the number of hidden units of the compression network.

On the other hand, Syntactic NNs can model stochastic and non-stochastic grammars. Learning is therefore a process of grammatical inference and recognition a process of parsing. Note that this has great generality; by varying the grammar we can encompass a wide range of pattern recognition models. The stochastic nets are properly probabilistic and are powerful discriminators; the non-stochastic are less powerful, but have straightforward silicon implementation with existing technology. Learning in syntactic nets may proceed supervised or unsupervised (Lucas, 1990).

Combined Classifiers Approaches

(Baltzakis, 2001) presents a different technique for off-line signature recognition and verification. The proposed confronts above mentioned BPN problems by reducing the training computation time (This is achieved because each neural network corresponds to only one signature owner) and the size of the neural networks used (The feature set is split to three different groups, i.e., global features, grid features and texture features.). For each one of these feature sets a special two stage Perceptron OCON (one-class-one-network) classification structure has been implemented. In the

first stage, the classifier combines the decision results of the neural networks and the Euclidean distance obtained using the three feature sets. The results of the first-stage classifier feed a second-stage radial base function (RBF) neural network structure, which makes the final decision.

To effectively verify skilled forgeries, a fuzzy neural network named *Pseudo Outer-Product based Fuzzy Neural Network* (POPFNN) is integrated into the signature verification system described in (Zhou, 1996). As a hybrid of fuzzy systems and neural networks, the POPFNN possesses many advantages such as high computational capability and learning ability when compared against other techniques used in signature verification systems. As hybrid intelligent systems, fuzzy NNs possess the advantages of both NNs and fuzzy rule-based systems and are particularly powerful in handling complex, non-linear and imprecise problems such as ASV. Besides, the membership functions and fuzzy rules identified in the POPFNN give more transparency to the decision making process. These advantages make the proposed fuzzy neural network driven signature verification system particularly powerful and robust even in dealing with skilled forgeries. In (Zhou, 1996), POPFNN operates in two fundamental modes, the learning mode and the classification mode. In the learning mode, a collection of training signature samples is used to train POPFNN. Feature vectors extracted from the training signature samples are utilized to initialize and adjust the parameters of POPFNN, including membership functions, fuzzy rules, and weights of the links. In the classification mode, POPFNN performs pure classification without self-modification. Feature vectors extracted from the unknown signatures are fed into POPFNN and the corresponding outputs are obtained at the output layer of POPFNN.

(Bromley, 1994) presents an algorithm based on a novel NN, called a "Siamese" neural network. This network has two input fields to compare two patterns and one output whose estate value corresponds to the similarity between the two patterns. During training the two sub-networks extract features from two signatures, while the joining neuron measures the distance between the two feature vectors. Training was carried out using a modified version of BP. All weights could be learnt, but the two sub-networks were constrained to have identical weights.

FUTURE TRENDS

Notwithstanding the enormous work carried out in the field of signature verification, several questions still remains unresolved. New solutions to these problems will determine the conditions under which the signature verification systems of the next future will be developed. The selection of the most suitable set of feature for a signer is one of the relevant open questions and the use of new approaches for classification still an open problem. Genetic algorithms (GA) have been recently used for this purpose (Xuhua, 1996). Another promising area of research concern multi-expert verification, which combine hard (Dimauro, 1997) and soft (Plamondon,1992) decision, based on parallel (Qi, 1995), serial (Cardot, 1991) or hybrid strategies(Cordella, 2000).

In the framework of a handwritten text recognition application, (Heutte, 2004) have developed a multiple agent system able to manage interaction between different contextual levels of handwriting interpretation. The EMAC (Hernoux, 1999) environment has been specified from constraints imposed by their handwriting interpretation system. This work presents this platform as help to implement specific collaboration or cooperation schemes between agents which bring out new trends in the automatic reading of handwritten texts and could be implemented for automatic signature verification systems.

(Balkrishana, 2007) recently presented a Colour Code Algorithm which deals with the recognition of the signature, as human operator generally make the work of signature recognition. Hence the algorithm simulates human behavior, to achieve perfection and skill through AI. The logic that decides the extent of validity of the signature must implement Artificial Intelligence Pattern recognition is the science that concerns the description or classification of measurements, usually based on underlying model. In future the system can be configured using Neural Networks and Fuzzy Rule base, where online training of recognition is possible.

A list of companies involved in signature verification systems production is given in (Kalenova, 2004), along with a short description of the products available. Although signature verification is not one of the safest biometric solutions, the use of it in business practices is still justified. Primarily due to the fact that the signature is a de facto mean of confirming the identity of the person, and therefore will provide a far less disruptive migration to an advanced technology than any other biometric can. Thus, signature verification has a very promising future.

CONCLUSION

Automatic signature verification is very attractive problem for researches. This article presents a review of approaches for Automatic Signature Verification using Neural Networks. The main aspects related to training process are discussed. Although some approaches have False Reject Rate and False Acceptance Rate ranging from 2% to 5%, systems developers cannot compare their results due to the lack of a widely accepted protocol for experimental tests, as well as the absence of large, public signature databases. A useful bibliography is also provided for interested readers.

REFERENCES

Abuhaiba, I.,(2007) *Offline Signature Verification Using Graph Matching*, Turkish Journal of Electrical Engineering & Computer Sciences. 1(4).

Balkrishana, V., (2007), A *Colour Code Algorithm for Signature Recognition*, Electronic Letters on Computer Vision and Image Analysis. 6(1), 1-12.

Baltzkis H, & Papamarkos N, (2001), *A new signature verification technique based on a two stage neural network classifier*, Engineering Application of Artificial Intelligence. 14, 95-103.

Bouletreau, V., Vincent, N., Sabourin, R. & Emptoz, H., (1998). *Handwriting and signature: one or two personality identifiers?.*, Proceedings Fourteenth International Conference on Pattern Recognition. 2, 1758-1760.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. & Shah, R., (1994), *Signature Verification using a "Siamese" Time Delay Neural Network*, Advances in Neural Information Processing Systems. 7(4), 669-688.

Cardot, H., Revenu, M., Victorri, B., & Revillet, M.J., (1991), *Cooperation de réseaux neuronaux pour*

l'autentification de signatures manuscrites, Proceedings of International Conference on Neuro-Nimes. 6, 737–744.

Cordella, L.P., Foggia, P., Sansone, C., Tortorella, F., & Vento, M., (2000), *A Cascaded Multiple Expert System for Verification*, Multiple Classifier Systems, editions J.Kittler and F.Roli, Springer. 1857, 330-339.

Dimauro, G., Impedovo, S., Pirlo, G., & Salzo, A., (1997), *A multi-expert signature verification system for bankcheck processing*, International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI). 11(5), 827-844

Drouhard, J.P., Sabourin, R., & Godbout, M., (1996), *A Neural Network Approach Off-line Signature Verification using directional PDF*, Pattern Recognition. 29(3). 415-424.

Ferrer, M.A., Alonso, J.B., & Travieso, C.M., (2005), *Offline geometric parameters for automatic signature verification using fixed-point arithmetic*, IEEE Transactions on Pattern Analysis and Machine Intelligence. 27(6), 993-997.

Fleming, M.K. & Cottrell, G.W., (1990), *Categorization of faces using unsupervised feature extraction*, Proceedings International Conference on Neural Networks II. 2, 65-70.

Guo J, Doermann D, & Rosenfeld A, (1997), *Local correspondence for detecting random forgeries*, Proceedings of the Fourth International Conference on Document Analysis and Recognition. 1, 319-323.

Heutte, L., Nosary, A., & Paquet. T., (2004), *A multiple agent architecture for handwritten text recognition*. Pattern Recognition. 37, 665-674.

Hernoux, C., (1999). *EMAC, Un environnement Multi-Agents à mémoire Collective*, Mémoire d'ingénieur, CNAM.

Hou W, Ye X, & Wang K, (2004). *A Survey of Off-line Signature Verification*, Proceedings International Conference on Intelligent Mechatronics and Automation.

Kalenova, D., (2004), *Personal Authentication Using Signature Recognition*, Department of Information Technology, Laboratory of Information Processing, Lappeenranta University of Technology.

Kohonen, t., (1977), *Associative Memory: A System Theoretic Approach*, Springer.

Lippmann, R. P., (1987) *An introduction to computing with neural nets*, IEEE ASSP Magazine. 4(2), 4-22.

Looney, C.G., (1997), *Pattern Recognition using Neural Networks: Theory and Algorithms for Engineers and Scientists*, Oxford University Press.

Lucas, S.M. & Damper, R.I. , (1990), *Signature verification with a syntactic neural ne*t, IJCNN International Joint Conference on Neural Networks.

Mighell, D. A., Wilkinson, T. S. & J. W. Goodman, (1989), *Backpropagation and Its Application to Handwritten Signature Verification*, Advances in Neural Information Processing Systems. 340-347.

O'Toole A.J., et al., (1993), *Low dimensional representation of faces in higher dimensions of the face space*, Journal of the Optical Society of America A., 10, 405-410.

Plamondon, R., Yergeau, P., & Brault, J., (1992), *A multi-level signature verification system*, From Pixels to Features III - Frontiers in Handwriting Recognition., S.Impedovo and J.C.Simon editions.

Plamondon, R. & Lorette, G., (1989), *Automatic signature verification and writer identification—The state of the art*, Pattern Recognition. 22(2), 107-131.

Qi Y Y, & Hunt B R, (1994), *Signature verification using global and grid features*, Pattern Recognition. 27(12), 1621-1629.

Qi, Y., & Hunt, B.R., (1995), *A multiresolution approach to computer verification of handwritten signatures*, IEEE Transaction on Image Processing. 4(6).

Sabourin R., & Drouhard J. P., (1992), *Offline signature verification using directional PDF and neural networks*, Proceedings 11th international conference on pattern recognition.

Sansone C, & Vento M, (2000), *Signature verification: increasing performance by a multistage system*, Pattern Analysis & Application. 3, 169-181.

Valentin, D., Abdi, H., O'Toole, A.J. & Cottrell, G.W. (1994)., *Connectionist Models of Face Processing: A Survey*, Pattern Recognition. 27(9), 1209-1230.

Xuhua, Y., Furuhashi, T., Obata, K., & Uchikawa, Y., (1996), *Selection of features for signature verification using the genetic algorithm*, Computers Ind. Eng.

Zhou, R.W. & Quek, C. , (1996), *An automatic fuzzy neural network driven signature verification system*, IEEE International Conference on Neural Networks.

KEY TERMS

Agent Based Mode: A specific individual based computational model for computer simulation extensively related to the theme in complex systems, Monte Carlo Method, multi agent systems, and evolutionary programming. The idea is to construct the computational devices (agents with some properties) and then, simulate them in parallel to model the real phenomena.

Automatic Signature Verification: A procedure that determine if a handwritten signature is genuine or a forgery, when a person claims for identity verification.

Backpropagation Algorithm: Learning algorithm of ANNs, based on minimising the error obtained from the comparison between the outputs that the network gives after the application of a set of network inputs and the outputs it should give (the desired outputs).

Feature Selection: The technique, commonly used in machine learning, of selecting a subset of relevant features for building robust learning models. Its objective is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data.

Fuzzy Logic: Derived from fuzzy set theory dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. It can be thought of as the application side of fuzzy set theory dealing with well thought out real world expert values for a complex problem.

Genetic Algorithms: A genetic algorithm is technique used for searching or programming. It is used in computing to find true or approximate solutions to optimization and search problems of various types and used as a function in evolutionary computation. Genetic algorithms are based on biological events. They mimic biological evolution.

Principal Component Analysis: A technique used to reduce multidimensional data sets to lower dimensions for analysis. PCA involves the computation of the eigenvalue decomposition of a data set, usually after mean centering the data for each attribute.

N

Neural/Fuzzy Computing Based on Lattice Theory

Vassilis G. Kaburlasos
Technological Educational Institution of Kavala, Greece

INTRODUCTION

Computational Intelligence (CI) consists of an evolving collection of methodologies often inspired from nature (Bonissone, Chen, Goebel & Khedkar, 1999, Fogel, 1999, Pedrycz, 1998). Two popular methodologies of CI include neural networks and fuzzy systems.

Lately, a unification was proposed in CI, at a "data level", based on lattice theory (Kaburlasos, 2006). More specifically, it was shown that several types of data including vectors of (fuzzy) numbers, (fuzzy) sets, 1D/2D (real) functions, graphs/trees, (strings of) symbols, etc. are partially (lattice)-ordered. In conclusion, a unified cross-fertilization was proposed for knowledge representation and modeling based on lattice theory with emphasis on clustering, classification, and regression applications (Kaburlasos, 2006).

Of particular interest in practice is the totally-ordered lattice $(\mathsf{R},\leq)$ of real numbers, which has emerged historically from the conventional measurement process of successive comparisons. It is known that $(\mathsf{R},\leq)$ gives rise to a hierarchy of lattices including the lattice $(\mathsf{F},\leq)$ of *fuzzy interval numbers*, or FINs for short (Papadakis & Kaburlasos, 2007).

This article shows extensions of two popular neural networks, i.e. *fuzzy-ARTMAP* (Carpenter, Grossberg, Markuzon, Reynolds & Rosen 1992) and *self-organizing map* (Kohonen, 1995), as well as an extension of conventional *fuzzy inference systems* (Mamdani & Assilian, 1975), based on FINs. Advantages of the aforementioned extensions include both a capacity to rigorously deal with nonnumeric input data and a capacity to introduce tunable nonlinearities. *Rule induction* is yet another advantage.

BACKGROUND

Lattice theory has been compiled by Birkhoff (Birkhoff, 1967). This section summarizes selected results regarding a Cartesian product lattice $(\mathsf{L},\leq)=(\mathsf{L}_1,\leq_1)\times\ldots\times(\mathsf{L}_\mathsf{N},\leq_\mathsf{N})$ of *constituent* lattices $(\mathsf{L}_i,\leq_i)$, i=1,…,N.

Given an *isomorphic* function $\theta_i\colon (\mathsf{L}_i,\leq_i)\to(\mathsf{L}_i,\leq_i)^\partial$ in a constituent lattice $(\mathsf{L}_i,\leq_i)$, i=1,…,N, where $(\mathsf{L}_i,\leq_i)^\partial \equiv (\mathsf{L}_i,\leq_i^\partial)$ denotes the *dual* (lattice) of lattice $(\mathsf{L}_i,\leq_i)$, then an isomorphic function $\theta\colon (\mathsf{L},\leq)\to(\mathsf{L},\leq)^\partial$ is given by $\theta(x_1,\ldots,x_\mathsf{N})=(\theta_1(x_1),\ldots,\theta_\mathsf{N}(x_\mathsf{N}))$.

Given a *positive valuation* function $v_i\colon (\mathsf{L}_i,\leq_i)\to\mathsf{R}$ in a constituent lattice $(\mathsf{L}_i,\leq_i)$, i=1,…,N then a positive valuation $v\colon(\mathsf{L},\leq)\to\mathsf{R}$ is given by $v(x_1,\ldots,x_\mathsf{N})=v_1(x_1)+\ldots+v_\mathsf{N}(x_\mathsf{N})$.

It is well-known that a positive valuation $v_i\colon(\mathsf{L}_i,\leq_i)\to\mathsf{R}$ in a lattice $(\mathsf{L}_i,\leq_i)$ implies a metric function $d_i\colon \mathsf{L}_i\times\mathsf{L}_i\to\mathsf{R}_0^+$ given by $d_i(a,b) = v_i(a\vee b) - v_i(a\wedge b)$.

Minkowski metrics $d_p\colon(\mathsf{L}_1,\leq_1)\times\ldots\times(\mathsf{L}_\mathsf{N},\leq_\mathsf{N})=(\mathsf{L},\leq)\to\mathsf{R}$ are given by

$$d_p(x,y) = \left[d_1^p(x_1,y_1)+\ldots+d_\mathsf{N}^p(x_\mathsf{N},y_\mathsf{N}) \right]^{1/p},$$

where

$x=(x_1,\ldots,x_\mathsf{N})$, $y=(y_1,\ldots,y_\mathsf{N})$, $p\in\mathsf{R}$.

An *interval* $[a,b]$ in a lattice $(\mathsf{L},\leq)$ is defined as the set $[a,b]\doteq\{x\in\mathsf{L}\colon a\leq x\leq b, a,b\in\mathsf{L}\}$. Let $\tau(\mathsf{L})$ denote the set of intervals in a lattice $(\mathsf{L},\leq)$. It turns out that $(\tau(\mathsf{L}),\leq)$ is a lattice, ordered by set inclusion.

Definition 1. The *size* $Z_p\colon \tau(\mathsf{L})\to\mathsf{R}_0^+$ of a lattice $(\mathsf{L},\leq)$ interval $[a,b]\in\tau(\mathsf{L})$, with respect to a positive valuation $v\colon(\mathsf{L},\leq)\to\mathsf{R}$, is defined as $Z_p([a,b])=d_p(a,b)$.

NEURAL/FUZZY COMPUTING BASED ON LATTICE THEORY

This section delineates modified extensions to a hierarchy of lattices stemming from the totally ordered lattice $(\mathsf{R},\leq)$ of real numbers. Then, it details the relevance of

novel mathematical tools. Next, based on the previous mathematical tools, this section presents extensions of ART/SOM/FIS. Finally, it discusses comparative advantages.

Modified Extensions in a Hierarchy of Lattices

Consider the product lattice $(\Delta,\leq) = (\mathsf{R}\times\mathsf{R},\leq^{\partial}\times\leq) = (\mathsf{R}\times\mathsf{R},\geq\times\leq)$ of *generalized intervals*. A *generalized interval* (element in Δ) will be denoted by $[a,b]$ and will be called *positive* (*negative*) for $a\leq b$ ($a>b$). The set of positive (negative) generalized intervals will be denoted by Δ_+ (Δ_-) − We remark that the set of *positive* generalized intervals is isomorphic to the set of *conventional intervals* in the set R of real numbers.

A *decreasing* function θ_{R}: $\mathsf{R}\to\mathsf{R}$ is an isomorphic function θ_{R}: $(\mathsf{R},\leq)\to(\mathsf{R},\leq)^{\partial}$; furthermore, a *strictly increasing* function v_{R}: $\mathsf{R}\to\mathsf{R}$ is a positive valuation v_{R}: $(\mathsf{R},\leq)\to\mathsf{R}$. Hence, function v_{Δ}: $(\Delta,\leq)\to\mathsf{R}$ given by $v_{\Delta}([a,b])=v_{\mathsf{R}}(\theta_{\mathsf{R}}(a))+v_{\mathsf{R}}(b)$ is a positive valuation in lattice $(\Delta,\leq)$. There follows a metric function d_{Δ}: $\Delta\times\Delta\to\mathsf{R}_0^+$ given by $d_{\Delta}([a,b],[c,d])=[v_{\mathsf{R}}(\theta_{\mathsf{R}}(a\wedge c))-v_{\mathsf{R}}(\theta_{\mathsf{R}}(a\vee c))] + [v_{\mathsf{R}}(b\vee d)-v_{\mathsf{R}}(b\wedge d)]$; in particular, for $\theta_{\mathsf{R}}(x)= -x$ and $v_{\mathsf{R}}(x)= x$ it follows $v_{\Delta}([a,b])= |a-c| + |b-d|$. Choosing *parametric* functions $\theta_{\mathsf{R}}(.)$ and $v_{\mathsf{R}}(.)$ there follow tunable nonlinearities in lattice $(\mathsf{R},\leq)$. Moreover, note that Δ is a *real linear space* with

- *addition* defined as $[a,b] + [c,d] = [a+c,b+d]$, and
- *multiplication* (by a real k) defined as $k[a,b] = [ka,kb]$.

It turns out that Δ_+ (as well as Δ_-) is *cone* in linear space Δ − Recall that a subset C of a linear space is called *cone* if for all $x\in C$ and $\lambda>0$, we have $\lambda x\in C$.

Definition 2. A *generalized interval number* (*GIN*) is a function f: $(0,1]\to\Delta$.

Let G denote the set of GINs. It follows that $(\mathsf{G},\leq)$ is a lattice, in particular $(\mathsf{G},\leq)$ is the Cartesian product of lattices $(\Delta,\leq)$. Moreover, G is a real linear space with

- *addition* defined as $(G_1 + G_2)(h) = G_1(h) + G_2(h)$, $h\in(0,1]$, and
- *multiplication* (by a real k) defined as $(kG)(h) = kG(h)$, $h\in(0,1]$.

We remark that the cardinality of set G equals $\aleph_1^{\aleph_1} = \left(2^{\aleph_0}\right)^{\aleph_1} = 2^{\aleph_0\aleph_1} = 2^{\aleph_1} = \aleph_2 > \aleph_1$, where $\aleph_1$ is the cardinality of the set R of real numbers.

Proposition 3. Consider metric(s) d_{Δ}: $\Delta\times\Delta\to\mathsf{R}_0^+$ in lattice $(\Delta,\leq)$. Let $G_1,G_2\in(\mathsf{G},\leq)$. Assuming that the following integral exists, a metric function d_{G}: $\mathsf{G}\times\mathsf{G}\to\mathsf{R}_0^+$ is given by

$$d_{\mathsf{G}}(G_1,G_2) = \int_0^1 d_{\Delta}(G_1(h),G_2(h))dh.$$

Our interest here focuses on the *sublattice* $(\mathsf{F},\leq)$ of lattice $(\mathsf{G},\leq)$, namely sublattice of *fuzzy interval numbers* (*FIN*s). A FIN is defined rigorously as follows.

Definition 4. A *fuzzy interval number* (*FIN*) F is a GIN such that either (1) both $F(h)\in\Delta_+$ and $h_1\leq h_2 \Rightarrow F(h_1)\geq F(h_2)$, for all $h\in(0,1]$ (*positive FIN*) or (2) there is a positive FIN P such that $F(h) = -P(h)$, for all $h\in(0,1]$ (*negative FIN*).

Let F_+ (F_-) denote the set of positive (negative) FINs. Note that both $\mathsf{F}_+\cup\mathsf{F}_- = \mathsf{F}$ and $\mathsf{F}_+\cap\mathsf{F}_-=\varnothing$ hold. Furthermore, F_+ (F_-) is a cone with cardinality $\aleph_1$ (Kaburlasos & Kehagias, 2006). The previous mathematical analysis may potentially produce useful techniques based on lattice vector theory (Vulikh, 1967). A *positive FIN* will simply be called "FIN". A FIN may admit different interpretations including a (fuzzy) number, an interval, and a cumulative distribution function.

Relevance of Novel Mathematical Tools

A fundamental mathematical result in *fuzzy set theory* is the "resolution identity theorem", which states that a fuzzy set can, equivalently, be represented either by its membership function or by its α-cuts (Zadeh, 1975). The aforementioned theorem has been given little attention in practice to date. However, some authors have capitalized on it by designing effective as well as efficient fuzzy inference systems (FIS) involving fuzzy numbers whose α-cuts are conventional closed intervals (Uehara & Fujise, 1993, Uehara & Hirota, 1998).

This work builds on the abovementioned mathematical result as follows. In the first place, we drop the possibilistic interpretation of a membership function. Then, we consider the corresponding "α-cuts representation".

Next, we consider the metric cone F_+^N of (positive) FINs. In conclusion, we propose extensions of established neural/fuzzy algorithms, including *ART* (adaptive resonance theory), *SOM* (self-organizing map), and *FIS* (fuzzy inference systems), in F_+^N (Kaburlasos, 2007). A novelty of this work is an improved mathematical notation, which emphasizes relevance with the aforementioned "resolution identity theorem".

An Extension of Fuzzy-ARTMAP

A fuzzy-ARTMAP extension, namely *fuzzy lattice reasoning* (*FLR*), is presented in this section based on a similarity measure (function) defined in the following.

Definition 5. A *similarity measure* in a set S is a function $\mu: S \times S \to (0,1]$, which satisfies the following conditions.

(S1) $\mu(a,b) = 1 \Leftrightarrow a = b$.
(S2) $\mu(a,b) = \mu(b,a)$.
(S3) $\dfrac{1}{\mu(a,b)} + \dfrac{1}{\mu(x,x)} \leq \dfrac{1}{\mu(a,x)} + \dfrac{1}{\mu(x,b)}$.

A similarity measure is defined based on a metric function next.

Proposition 6. If function $d: S \times S \to \mathsf{R}_0^+$ is a metric then function $\mu: S \times S \to (0,1]$ given by $\mu(a,b) = 1/[1+d(a,b)]$ is a similarity measure.

FLR for Training

FLR-0: A set $RB = \{(u_1,C_1),\dots,(u_L,C_L)\}$ is given, where $u_l \in \mathsf{F}_+^N$ and $C_l \in \mathsf{C}$, $l=1,\dots,L$ is a class label in the finite set C.

FLR-1: Present the next input pair $(x_i,K_i) \in \mathsf{F}_+^N \times \mathsf{C}$, $i=1,\dots,n$ to the initially "set" RB.

FLR-2: If no more pairs are "set" in RB then store input pair (x_i,K_i) in the RB; $L \leftarrow L+1$; goto step FLR-1.
Else, compute the similarity $\mu(x_i,u_l)$ of input $x_i \in \mathsf{F}_+^N$ with a "set" element $u_l \in \mathsf{F}_+^N$, $l=1,\dots,L$ in RB.

FLR-3: Competition among the "set" pairs in the RB: Winner is pair (u_J,C_J) such that $J \doteq arg \max\limits_{l \in \{1,\dots,L\}} \mu(x_i,u_l)$. In case of multiple winners, choose the one with the smallest size $Z_1(.)$.

FLR-4: *Assimilation Condition*: Both (1) size $Z_1(x_i \vee u_J)$ is less than a user-defined threshold size Z_{crit}, and (2) $K_i = C_J$.

FLR-5: If the *Assimilation Condition* is not satisfied then "reset" the winner pair (u_J,C_J); goto step FLR-2.
Else, replace the winner u_J by the join-interval $x_i \vee u_J$; goto step FLR-1.

The corresponding testing phase is carried out by winner-take-all competition based on the similarity measure function $\mu(.,.)$.

An Extension of SOM

A straightforward SOM extension, namely *granular SOM* (*grSOM*), is presented in this section in cone F_+^N.

grSOM for Training

GR-0: The user defines the size L of a $L \times L$ grid of neurons. Each neuron can store both a N-dimensional FIN $W_{i,j} \in \mathsf{F}_+^N$, $i,j \in \{1,\dots,L\}$ and a class label $C_{i,j} \in \mathsf{C}$, where C is a finite set. Initially all neurons are *uncommitted*.

GR-1: Memorize the first training data pair $(x_1,K_1) \in \mathsf{F}_+^N \times \mathsf{C}$ by committing, randomly, a neuron in the $L \times L$ grid.
Repeat the following steps a user-defined number N_{epochs} of epochs.

GR-2: For each training datum $(x_k,K_k) \in \mathsf{F}_+^N \times \mathsf{C}$, $k=1,\dots,n$ "reset" all $L \times L$ grid neurons. Then carry out the following computations.

GR-3: Calculate the Minkowski metric distance $d_1(x_k,W_{i,j})$ between x_k and *committed* neurons $W_{i,j}$, $i,j \in \{1,\dots,L\}$.

GR-4: Competition among the "set" (and, *committed*) neurons in the $L \times L$ grid: Winner is neuron (I,J) whose weight $W_{I,J}$ is the nearest to x_k, that is $(I,J) \doteq arg \min\limits_{i,j \in \{1,\dots,L\}} d_1(x_k,W_{i,j})$.

GR-5: *Assimilation Condition*: Both (1) Vector $W_{i,j}$ is in the neighborhood of vector $W_{I,J}$ on the $L \times L$ grid, and (2) $C_{I,J} = K_k$.

GR-6: If the *Assimilation Condition* is satisfied then compute a new value $W'_{i,j}$ as

N

$$W'_{i,j} \doteq \left[1 - \frac{h(k)}{1 + d_1(W_{I,J}, W_{i,j})} \right] W_{i,j} + \frac{h(k)}{1 + d_1(W_{I,J}, W_{i,j})} x_k$$

Else, "reset" the winner (I,J); goto GR-4.

GR-7: If all the $L \times L$ neurons are "reset" then commit an *uncommitted* neuron from the grid, and memorize the current training datum (x_k, K_k).

If there are no more *uncommitted* neurons then increase L by one.

The corresponding testing phase is carried out by winner-take-all competition based on the Minkowski metric $d_1(.,.)$.

An Extension of FIS

The basic idea towards novel FIS analysis and design is to employ a similarity measure function $\mu(X, A_i) = 1/[1 + d(X, A_i)]$, where $X, A_i \in \mathsf{F}_+^N$, as a fuzzy membership function regarding a rule $R_i: A_i \rightarrow C_i$, where $A_i \in \mathsf{F}_+^N$, $C_i \in \mathsf{F}_+^M$, i=1,...,L (Kaburlasos & Kehagias, 2007). Advantages are presented in the following.

Comparative Advantages

First, an important advantage of the mathematical tools above is that the proposed ART/SOM/FIS extensions can handle, in any combination, numeric and/or non-numeric data, the latter include fuzzy numbers, intervals, and cumulative distribution functions.

Second, we can employ parametric decreasing (increasing) functions $\theta_R: \mathsf{R} \rightarrow \mathsf{R}$ ($v_R: \mathsf{R} \rightarrow \mathsf{R}$) in a data dimension, where the function parameters can be estimated/tuned optimally towards improving performance.

Third, the proposed ART/SOM/FIS extensions can induce descriptive decision-making knowledge (i.e. rules) from the training data.

Fourth, regarding the FLR, note that a *similarity measure* function $\mu(.,.)$ can effectively replace an *inclusion measure* function $\sigma(.,.)$ – Recall that the latter (function) had replaced both of fuzzy-ARTMAP's *Choice* (*Weber*) function and *Match* function (Kaburlasos & Petridis, 2000, Kaburlasos, Athanasiadis & Mitkas, 2007). The reason behind the aforementioned "effective" replacement is that an *inclusion measure* $\sigma(A, B)$, or $\sigma(B, A)$, considers mainly one of $A, B \in \mathsf{F}_+^N$; whereas,

a *similarity measure* $\mu(A, B)$ considers both $A, B \in \mathsf{F}_+^N$ based on their corresponding metric distance.

Fifth, regarding the proposed SOM extension, note that this work carries out computations in the cone F_+ of FINs for faster data processing compared to a previous version of grSOM (Kaburlasos & Papadakis, 2006).

Sixth, regarding the proposed FIS, novel advantages include a capacity to generalize beyond a fuzzy rule's support. The latter implies, potentially, an alleviation of the "curse of dimensionality" problem regarding the number of rules.

FUTURE TRENDS

Data-processing of FINs by multiplayer perceptrons is straightforward, as described in (Kaburlasos & Christoforidis, 2006), and it will be pursued in future work.

CONCLUSION

This article has presented novel mathematical tools for unified analysis and design of neural/fuzzy systems. We built on fuzzy set theory's "resolution identity theorem". Nevertheless, in the first place, we dropped the possibilistic interpretation of a membership function. Then, we considered the corresponding "α-cuts representation". Our interest focused on fuzzy interval numbers, or FINs for short, which can represent (fuzzy) numbers, intervals, and cumulative distribution functions. Based on lattice theory, we showed that the space of FINs is a metric cone. In conclusion, this works opens up the possibility to design FIN-to-FIN maps implementable on neural/fuzzy architectures including also tunable nonlinearities.

REFERENCES

Birkhoff, G. (1967). Lattice Theory. Providence, RI: AMS, Colloquium Publications, 25.

Bonissone, P.P., Chen, Y.T., Goebel, K., & Khedkar, P.S. (1999) Hybrid Soft Computing Systems: Industrial and Commercial Applications. Proc IEEE, (87) 9, 1641-1667.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., & Rosen, D.B. (1992) Fuzzy ARTMAP:

A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. IEEE Transactions on Neural Networks, (3) 5, 698-713.

Fogel, D.B. (1999). Evolutionary Computation: Toward a New Philosophy of Machine Intelligence (2nd ed.). Piscataway, NJ: IEEE Press.

Kaburlasos, V.G. (2006). Towards a Unified Modeling and Knowledge Representation Based on Lattice Theory − Computational Intelligence and Soft Computing Applications. Heidelberg, Germany: Springer, series: Studies in Computational Intelligence, vol. 27.

Kaburlasos, V.G. (2007). Unified Analysis and Design of ART/SOM Neural Networks and Fuzzy Inference Systems Based on Lattice Theory. Computational and Ambient Intelligence, Sandoval, F., Prieto, A., Cabestany, J., Graña, M. editors. Heidelberg, Germany: Springer-Verlag, series: LNCS, vol. 4507, pp. 80-93.

Kaburlasos, V.G., & Christoforidis, A. (2006). Granular Auto-Regressive Moving Average (grARMA) Model for Predicting a Distribution From Other Distributions. Real-World Applications. Proceedings of the World Congress on Computational Intelligence (WCCI) 2006, FUZZ-IEEE Program, pp. 791-796.

Kaburlasos, V.G., & Kehagias, A. (2006). Novel Fuzzy Inference System (FIS) Analysis and Design Based on Lattice Theory. Part I: Working Principles. International Journal of General Systems, (35) 1, 45-67.

Kaburlasos, V.G., & Kehagias, A. (2007). Novel Fuzzy Inference System (FIS) Analysis and Design Based on Lattice Theory. IEEE Trans. Fuzzy Systems, (15) 2, 243-260.

Kaburlasos, V.G., & Papadakis, S.E. (2006). Granular Self-Organizing Map (grSOM) for Structure Identification. Neural Networks, (19) 5, 623-643.

Kaburlasos, V.G., & Petridis, V. (2000). Fuzzy Lattice Neurocomputing (FLN) Models. Neural Networks, (13) 10, 1145-1170.

Kaburlasos, V.G., Athanasiadis, I.N., & Mitkas, P.A. (2007). Fuzzy Lattice Reasoning (FLR) Classifier and its Application for Ambient Ozone Estimation. International Journal of Approximate Reasoning, (45) 1, 152-188.

Kohonen, T. (1995). Self-Organizing Maps. Berlin, Germany: Springer.

Mamdani, E.H., & Assilian, S. (1975). An Experiment in Linguistic Synthesis With a Fuzzy Logic Controller. International Journal of Man-Machine Studies, (7), 1-13.

Papadakis, S.E., & Kaburlasos, V.G. (2007). Induction of Classification Rules From Histograms. Joint Conference on Information Sciences (JCIS), Proceedings of the 8th International Conference on Natural Computing(NC), pp. 1646-1652.

Pedrycz, W. (1998). Computational Intelligence − An Introduction. Boca Raton, FL: CRC Press.

Uehara, K., & Fujise, M. (1993). Fuzzy Inference Based on Families of α-level sets. IEEE Transactions on Fuzzy Systems, (1) 2, 111-124.

Uehara, K., & Hirota, K. (1998). Parallel and Multistage Fuzzy Inference Based on Familes of α-level sets. Information Sciences, (106) 1-2, 159-195.

Vulikh, B.Z. (1967). Introduction to the Theory of Partially Ordered Vector Spaces. Gronigen: Wolters-Noordhoff Scientific Publications, XV.

Zadeh, L.A. (1975) The Concept of a Linguistic Truth Variable and its Application to Approximate Reasoning –I, II, III. Information Sciences, (8) 3, 199-249; (8) 4, 301-357; (9) 1, 43-80.

KEY TERMS

ART: ART stands for Adaptive Resonance Theory. That is a biologically inspired neural paradigm for, originally, clustering binary patterns. An analog pattern version of ART, namely *fuzzy-ART*, is applicable in the unit hypercube. The corresponding neural network for classification is called *fuzzy-ARTMAP*.

Dual (Lattice): Given a lattice $(L, \leq)$, its *dual* lattice, symbolically $(L, \leq)^{\partial}$ or $(L, \leq^{\partial}) \equiv (L, \geq)$, is a lattice with the inverse order relation $(\geq)$.

FIS: FIS stands for Fuzzy Inference System. That is an architecture for reasoning involving fuzzy sets (typically fuzzy numbers) based of fuzzy logic.

Isomorhic (Function): Given two lattices $(L_1, \leq_1)$ and $(L_2, \leq_2)$, an *isomorphic* function is a bijective (one-to-one) function $\varphi: (L_1, \leq_1) \rightarrow (L_2, \leq_2)$ such that $x \leq y \Leftrightarrow \varphi(x) \leq \varphi(y)$.

Lattice: A *lattice* is a poset $(L, \leq)$ any two of whose elements have both a greatest lower bound (g.l.b.), denoted by $x \wedge y$, and a least upper bound (l.u.b.), denoted by $x \vee y$.

Poset: A *partially ordered set* (or, *poset*, for short) is a pair $(P, \leq)$, where P is a set and $\leq$ is an order relation on P. The latter (relation) by definition satisfies (1) $x \leq x$, (2) $x \leq y$ and $y \leq x \Rightarrow x = y$, and (3) $x \leq y$ and $y \leq z \Rightarrow x \leq z$.

Positive Valuation (Function): Given a lattice $(L, \leq)$, a *positive valuation* is a function $v: (L, \leq) \rightarrow$ R, which satisfies both $v(x) + v(y) = v(x \wedge y) + v(x \vee y)$ and $x < y \Rightarrow v(x) < v(y)$.

Rule Induction: Process of learning, from cases or instances, if-then rule relationships that consist of an antecedent (if-part, defining the preconditions or coverage of the rule) and a consequent (then-part, stating a classification, prediction, or other expression of a property that holds for cases defined in the antecedent).

SOM: SOM stands for Self-Organizing Map. That is a biologically inspired neural paradigm for clustering analog patterns. SOM is often used for visualization of nonlinear relations of multi-dimensional data.

Subattice: A *sublattice* $(S, \leq)$ of a lattice $(L, \leq)$ is another lattice such that both $S \subseteq L$ and $x, y \in S \Rightarrow x \wedge y, x \vee y \in S$.

A New Self–Organizing Map for Dissimilarity Data

Tien Ho-Phuoc
GIPSA-lab, France

Anne Guerin-Dugue
GIPSA-lab, France

INTRODUCTION

The **Self-Organizing Map** (Kohonen, 1997) is an effective and a very popular tool for data clustering and visualization. With this method, the input samples are projected into a low dimension space while preserving their topology. The samples are described by a set of features. The input space is generally a high dimensional space R^d. 2D or 3D maps are very often used for visualization in a low dimension space (2 or 3).

For many applications, usually in psychology, biology, genetic, image and signal processing, such vector description is not available; only pair-wise **dissimilarity** data is provided. For instance, applications in Text Mining or ADN exploration are very important in this field and the observations are usually described through their proximities expressed by the "Levenshtein", or "String Edit" distances (Levenshtein, 1966). The first approach consists of the transformation of a dissimilarity matrix into a true **Euclidean distance** matrix. A straightforward strategy is to use "Multidimensional Scaling" techniques (Borg & Groenen, 1997) to provide a feature space. So, the initial vector SOM algorithm can be naturally used. If this transformation involves great distortions, the initial vector model for SOM is no longer valid, and the analysis of dissimilarity data requires specific techniques (Jain & Dubes, 1988; Van Cutsem, 1994) and Dissimilarity Self Organizing Map (DSOM) is a new one.

Consequently, adaptation of the **Self-Organizing Map** (SOM) to dissimilarity data is of a growing interest. During this last decade, different propositions emerged to extend the vector SOM model to pair-wise dissimilarity data. The main motivation is to cope with large proximity databases for data mining. In this article, we present a new adaptation of the SOM algorithm which is compared with two existing ones.

BACKGROUND

Basically, there are two main approaches to the SOM extension dealing with **dissimilarity** data. The first one uses a probabilistic framework, as for example in Graepel & Obermayer (1999) where a topographic mapping of proximity is derived by simulated annealing. The second approach uses directly the initial SOM framework to adapt the two usual steps (affectation, representation) to dissimilarity data, as for example in Kohonen & Somervuo (1998, 2002), in El Golli, Conan-Guez & Rossi (2004), and in Ambroise & Govaert, (1996).

Our work is inspired by this last approach and we have compared our proposal to the algorithms proposed by Kohonen (Kohonen & Somervuo, 1998) (Kohonen & Somervuo, 2002) and by El Golli et al. (El Golli, Conan-Guez & Rossi, 2004). Three metrics for quality estimate (quantization and neighborhood) are used for comparison. Numerical experiments on artificial and real data show the quality of the algorithm. The strong point of the proposed algorithm comes from a more accurate prototype estimate which is one of the most difficult parts of Dissimilarity SOM algorithms.

The major difficulty of the DSOM is the constraint on the output data representation. For (vector) SOM algorithm, there is a latent data model for each output **prototype** (a spherical distribution whose the prototype is the barycentre). For DSOM, there is no data model for each output prototype. One referent observation is explicitly associated to each output prototype instead of its tuning by the barycentre processing. This referent is usually chosen among the input observations at the end of an optimization process. Consequently, several prototypes can unfortunately share the same referent and these collisions provide great distortions in the output map. To avoid this difficulty, we propose here

an implicit referent for each prototype which is adapted during training iterations. So there is no collision during learning phase and consequently, the projection quality is greatly enhanced.

ADAPTATION OF SOM FOR DISSIMILARITY DATA

This article presents a new DSOM algorithm for dissimilarity data. We will first present DSOM algorithms which have been directly derived from the initial SOM framework. In the next parts, we will present in detail our proposed algorithm and some experiments to show its effectiveness in comparison with the other DSOM algorithms.

Description of DSOM Algorithms

Basically the starting point of the DSOM algorithm is the "batch" algorithm of the initial vector SOM. Let us recall this "batch" algorithm. At each iteration, the entire dataset is presented. We consider a dataset X of N observations, $X = \{o_i, i = 1..N\}$. The SOM is configured with C nodes (neurons) a priori interconnected on the output map where $\delta(c,l)$ is the distance between the nodes c and l. At iteration t, each node is represented by a prototype ω_c^t in the input space. After an initialization step, an affectation step and a representation step are sequentially processed at each iteration. The role of the former is to assign to each observation o_i, the best matching unit ω_{c*}^t, according to the **Euclidean distance**. The affectation function is:

$$c* = Arg\left[\underset{c}{Min}\left(d^2(o_i, \omega_c)\right)\right] \tag{1}$$

Thus, a partition of the whole dataset is realized. In the latter, the prototype ω_c is adjusted to represent each **partition** X_c as well as possible. This prototype is computed as the weighted average of the input samples. The weights are evaluated through the **neighborhood function** $h^T(.)$ which is a non-increasing function of the distance on the map and controlled by a radius parameter $T(t)$ decreasing with time. At the end, the prototype ω_c is the gravity centre of the partition X_c.

This representation step cannot be directly transposed to dissimilarity data. An alternative implementation is to approximate these C prototypes by referent observations belonging to the initial dataset X. Then, this step becomes very time-consuming: all the input observations are candidate and must be evaluated. Some strategies to reduce the computation time have been proposed (Conan-Guez, Rossi & El Golli, 2006).

Let us notice $D = [d_{ij}]$ $i, j = 1..N$, the dissimilarity data. These dissimilarities describe a non metric space. However, for all the DSOM algorithms, we consider symmetric dissimilarities.

For the DSOM proposed by Kohonen, each **prototype** will be represented by one referent observation, $\omega_c = o_{r(c)}$. During the initialisation step, C observations in the input dataset are randomly assigned to the prototypes. For the affectation step, the affectation function simply uses the input dissimilarity data. Each observation is assigned to the nearest prototype:

$$f(i) = Arg\left[\underset{c}{Min}\left(d_{ir(c)}\right)\right] = Arg\left[\underset{c}{Min}\left(d\left(o_i, o_{r(c)}\right)\right)\right] \tag{2}$$

For the representation step, a new observation $o_{r(c)}$ is assigned to the prototype ω_c minimizing the following cost function:

$$r(c) = Arg\left[\underset{j}{Min}\left(E(c, j)\right)\right] \tag{3}$$

where $E(c, j)$ is the weighted local distortion if o_j is the referent of the prototype ω_c:

$$E(c, j) = \sum_{o_i \in X} h^T\left(\delta\left(c, f(i)\right)\right)d^2\left(o_j, o_i\right) \tag{4}$$

The global cost function which is then minimized is the global distortion over all the prototypes:

$$E_g = \sum_{c=1}^{C} E(c, r(c)) \tag{5}$$

For the representation step, different variants are possible. The neighborhood function in Eq. (4) can be simply integrated on the neighborhood of the prototype (the search is realized over the union of the partitions inside an output neighborhood) and not on the weighted dissimilarities. It is the "set Mean search". Also, the exponent '2' in Eq. (4) can be omitted: it is the "set **Median** search".

Different prototypes can share the same referent (collision) when the search of the referent observa-

tions is limited to the input observations. So there is ambiguity for the affection step in the next iteration. This is the major difficulty of this approach. In some applications, for instance for symbol string organization, it is possible to search the "Median" or "Mean" outside the initial set: the referents in DSOM are not necessarily represented by elements belonging to the input space. But this optimization is an NP-hard problem. See Martínez, Juan & Casacuberta (2001) for a comparison of different strategies.

In El Golli, Conan-Guez & Rossi (2004), El Golli et al. propose a slightly different approach. The theoretical interest of this approach is the possibility to represent a prototype by more than one referent observation ($q \geq 1$). This allows to take into account a more complex latent data structure (multimodal distribution for instance) for each partition. Unfortunately in practice, it is difficult to choose the number (q) of referents by prototype and the optimization step becomes even more time-consuming. Let us describe here the algorithm for $q = 1$.

For the affectation step, a distance between an observation and a prototype is defined in Eq. (6). When the **neighborhood** is decreasing, this distance converges towards the initial dissimilarity. The representation step is the same as previously. With convergence, these two algorithms are similar.

$$D^T\left(o_i, \omega_c\right) = \sum_{l=1}^{C} h^T\left(\delta\left(c, l\right)\right) . d^2\left(o_i, o_{r(l)}\right) \tag{6}$$

Ambroise & Govaert (1996) proposes a different approach inspired from SEM (Stochastic Expectation Maximization) algorithm. The representation step is a "set Median search". The assignation step uses a stochastic process to affect each observation to the prototypes by a multinomial distribution (the proportions depend on the neighbourhood function and the affectation to the prototypes).

Description of the Proposed DSOM Algorithm

As explained previously, the difficulty is the representation step due to the lack of data model. The set of referent candidates is finite and distortions occur if several prototypes share the same referent. To overcome this situation, we propose an implicit representation step. Let us remark, during training and until convergence, a referent observation is only used to define the distance between a prototype and an input observation. So, we will define a proximity measure $D^T(o_i, \omega_c)$ without explicit referent to the prototype ω_c. The representation phase will simply adapt this proximity considering the new partition of the observations and the update of the neighborhood function. This simple implementation has a counterpart: it is necessary to define a data model based on latent Euclidean assumptions.

Let us consider a set X of vector samples, $X = \{x_i, i = 1..N, x_i \in R^d\}$. Let g be the gravity centre of X and $I(X)$, its **inertia**:

$$I(X) = \frac{1}{N} \sum_{x_i \in X} d^2(g, x_i)$$

.

All the samples have the same uniform weight ($\frac{1}{N}$). The inertia with respect to any observation e is then defined and decomposed thanks to the Huygens theorem:

$$I(X, e) = \frac{1}{N} \sum_{x_i \in X} d^2(e, x_i) = d^2(g, e) + I(X) \tag{7}$$

Moreover, $I(X)$ can be computed by considering all the distances $d(x_i, x_j)$:

$$I(X) = \frac{1}{N^2} \sum_{x_i \in X} \sum_{x_j \in X, j>i} d^2(x_i, x_j) \tag{8}$$

Thus, with Euclidean hypothesis, there is no need to know the gravity centre g, for computing the distance of any observation e to this gravity centre: $d^2(g,e) = I(X,e) - I(X)$.

We apply this principle to dissimilarity data. The input data is noticed o_i instead of x_i for vector data. The same formula is generalized to non uniform weighted observations. Let us consider one partition X_c associated to the **prototype** ω_c after the affectation step. The proximity between an observation o_i and the prototype ω_c is then defined by using the weights $m_{j/c}$ for each observation o_j given the prototype ω_c. The **inertia** $I(X_c)$ is evaluated over all the weighted dissimilarities:

$$D^T(o_i, \omega_c) = I(X_c, o_i) - I(X_c) = \sum_{o_j \in X} m_{j/c} d^2(o_i, o_j) - I(X_c)$$

$$m_{j/c} = \frac{h^T(\delta(c, f(j)))}{\sum_{l=1}^{N} h^T(\delta(c, f(l)))}, \sum_{j=1}^{N} m_{j/c} = 1$$

$$(9)$$

$$I(X_c) = \sum_{o_i \in X} \sum_{o_j \in X, j>i} m_{i/c} . m_{j/c} d^2(o_i, o_j)$$

$$(10)$$

Therefore, the algorithm is the following:

- Initialization step: Having an initial **partition**, X_c, $c = 1..C$, with for instance, an affectation from an initial random referent observation set.
- Representation step: For all prototypes ω_c and observations o_i, compute the weights $m_{i/c}$ in Eq. (9) and the inertia $I(X_c)$ in Eq. (10), update the neighborhood function for the next iteration.
- Affectation step: Affect each observation to a prototype $\omega_{f(i)}$ according to the minimum distance in Eq. (9):

$$f(i) = Arg\left[\underset{c}{Min}\left(D^T(o_i, \omega_c)\right)\right].$$

The representation step and affectation step are sequentially computed up to convergence. The training parameters for the decreasing neighborhood function follow the usual recommendations for SOM algorithms: fast, then slow decrease (http://www.cis.hut.fi/projects/somtoolbox/documentation/).

With convergence, if necessary for visualization of the final map, a referent observation can be associated to each prototype according to a "set Mean search" (or set Median) or a "Mean search" (or Median), for instance.

In the following, we will compare three DSOM respectively called DSOM(K), DSOM(EG) and DSOM for our proposal. To compare the "set Mean" and "set Median" approaches for the three algorithms, $d^2(o_i, o_j)$ will be substituted by $d^\gamma(o_i, o_j)$: "set Median" corresponds to $\gamma = 1$ and "set Mean" to $\gamma = 2$. Different power values γ will be also tested. Other transformations may be applied to a dissimilarity matrix to transform it into a distance matrix, such as adding a constant, or combining the both (Joly & le Calvé, 1994). The "adding constant" method provides great distortions in the

initial dissimilarity data. Our experiments confirm it. The "power" method gives better results.

Concerning the computation time, these DSOM algorithms are equivalent, but the reasons differ. For DSOM(K) and DSOM(EG), the representation step is the most time-consuming one due to optimization for each referent. With our proposal, this optimization is implicit, but this step remains time-consuming because of the computation of the weights $m_{i/c}$ and inertia $I(X_c)$.

Methodology Description of the Experiment

To evaluate the 3 DSOM algorithms, two metrics will be used. The first one is the classical quantization error (E_g). The second one concerns topology preservation. Among existing criteria, we have chosen two measures in Eq. (11) which are compatible with **dissimilarity** data: the "trustworthiness" (M_1) and the "continuity" (M_2) (Venna & Kaski, 2001). The **trustworthiness** relates to the error provided by new observations in an output neighborhood while they are not in the input neighborhood; conversely for the **continuity**. M_1 and M_2 are evaluated in function of the number (k) of the nearest neighbors and normalized between 0 and 1. For visualization according to Venna & Kaski, the trustworthiness is more important than the continuity. The more $M_1(k)$ and $M_2(k)$ are large, the better the projection quality is. We compute also the integrated $M_i(k)$ until a neighborhood with 10% of the whole samples: these values ($\overline{M_i}$) measure the quality of the local topology preservation.

$$M_1(k) = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{o_j \in U_k(o_i)} (r(o_i, o_j) - k)$$

$$M_2(k) = 1 - \frac{2}{Nk(2N - 3k - 1)} \sum_{i=1}^{N} \sum_{o_j \in V_k(o_i)} (\hat{r}(o_i, o_j) - k)$$

$$(11)$$

With $C_k(o_i)$, $\hat{C}_k(o_i)$ sets of k first neighbors of o_i in the input space, output space;

$$U_k(o_i) = \left\{ o_j \mid o_j \in \hat{C}_k(o_i) \wedge o_j \notin C_k(o_i) \right\},$$

$$V_k(o_i) = \left\{ o_j \mid o_j \notin \hat{C}_k(o_i) \wedge o_j \in C_k(o_i) \right\},$$

$r(o_i, o_j)$, $\hat{r}\left(o_i, o_j\right)$ ranks of o_j in the neighbourhood of o_i in the input space, output space.

Three databases are used. The first one is an artificial dataset: 100 uniform samples in R^2, dissimilarity data is the exact Euclidean distance, the configuration parameter γ is set to 2. The second one is the "Chicken Silhouette" (http://algoval.essex.ac.uk:8080/data/sequence/chicken/chicken.tgz). This data consists of 446 samples (binary images of chicken parts) categorized in 5 classes. The distance matrix is calculated according to "AngleCostFunction" (Barbara Spillmann, 2004) based on the local orientation of the sample contours. The third dataset is larger. It is extracted from the SCOWL word lists (http://wordlist.sourceforge.net/). After some reduction of plural and possessive forms from a small English dictionary, the dataset consists of 2000 words. The Levenshtein distance (Levenshtein, 1966) is then used to calculate the pair-wise dissimilarities.

Results

On the artificial dataset, the performances of the three algorithms are very similar (Table 1). With a vector SOM, the results are identical. The map is a hexagonal one with a grid of 5x5 neurons.

As expected, the behaviour of the three algorithms differs with the real datasets. With the "Chicken" databases, the map is a hexagonal one with a grid of 7x7 neurons. DSOM presents the best topology preservation according to $M_1(k)$ and $M_2(k)$ (Fig. 1.a), and the best compromise between quantization and topology preservation (Table 2). While varying γ, we observe an evolution of these criteria. We notice that each algorithm exhibits a different value for the optimal power γ: γ = 1 for DSOM(K), γ = 1.5 for DSOM(EG), γ = 3 for DSOM. However, γ = 1 can be considered as the best compromise for the three algorithms and will be used

Table 1. *Comparison of the quantization quality (E_g) and topology preservation ($\overline{M_1}, \overline{M_2}$)*

Artificial, γ = 2	DSOM(K)	DSOM(EG)	DSOM
E_g	0.0063	0.0067	0.0063
$\overline{M_1}$	0.9892	0.9848	0.9855
$\overline{M_2}$	0.9791	0.9777	0.9804

Table 2. *Comparison of the quantization quality (E_g) and topology preservation ($\overline{M_1}, \overline{M_2}$)*

Chicken, γ = 1	DSOM(K)	DSOM(EG)	DSOM
E_g	11.7183	12.0817	11.7966
$\overline{M_1}$	0.8923	0.9040	0.9360
$\overline{M_2}$	0.8320	0.8083	0.8880

Figure 1. (a) Chicken database: Evolution of M₁(k) and M₂(k) with γ =1, (b) SCOWL database: Evolution of $\overline{M}_1$ and $\overline{M}_2$ for different values of the power γ

(a)

(b)
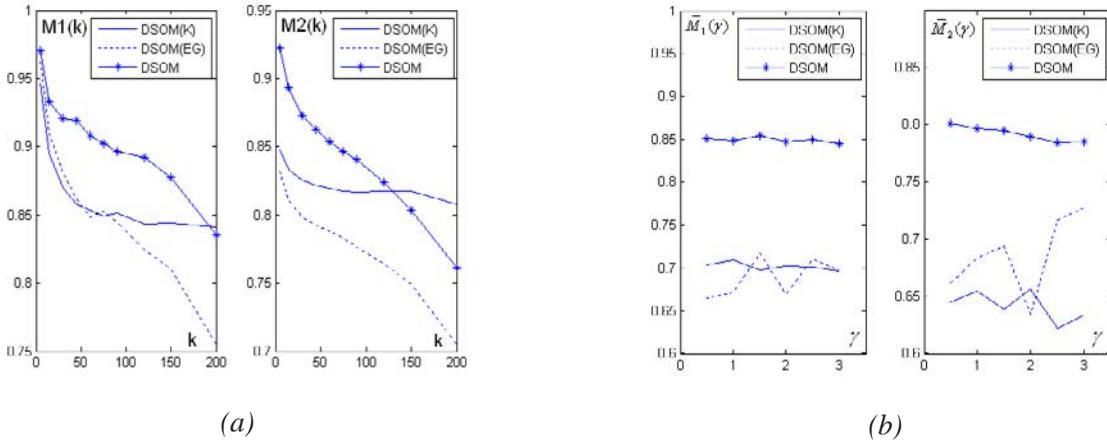
Figure 2. Chicken database: prototypes of the neurons for DSOM. Each color corresponds to one of five classes of chicken parts: wing, back, drumstick, thigh and back, and breast.
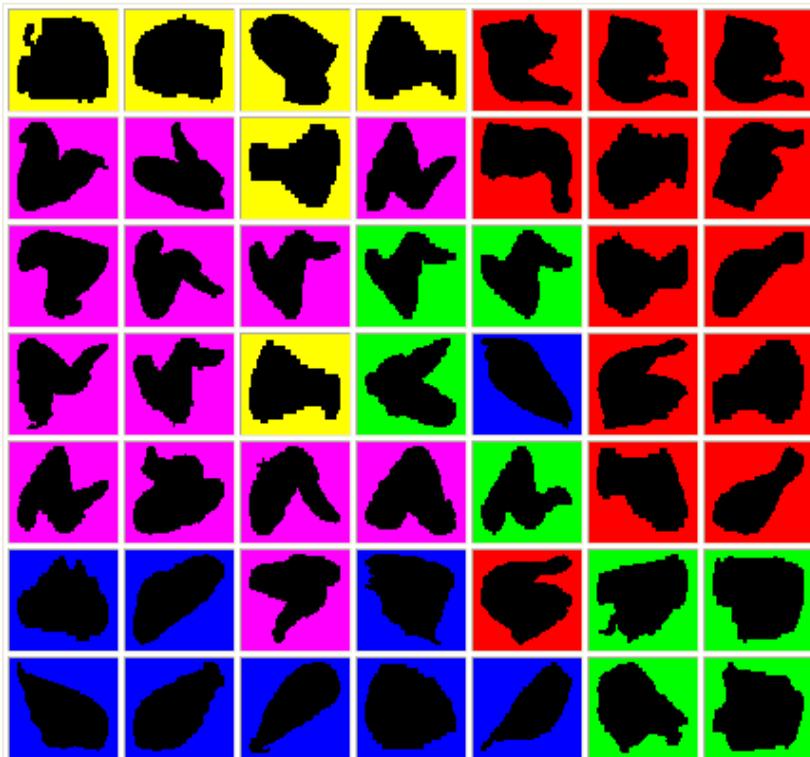
Figure 3. SCOWL database: Part of the final map. At the end, the referents are assigned with a "set Median search". For the particularity of referent 117, see the text.
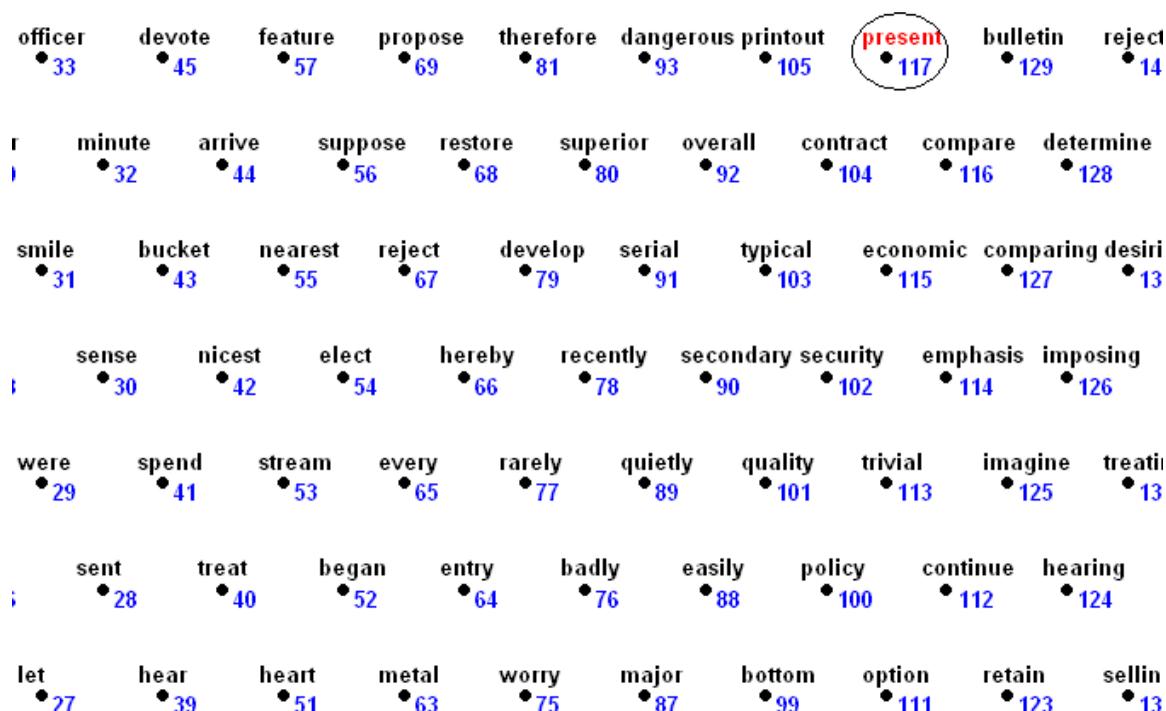
to present the results. Figure 2 show the prototypes of all the nodes for DSOM. The neighbor nodes have the similar prototypes. The map is organized to respect the data clustered into the 5 classes as well as possible.

For the third dataset, the hexagonal map is used with the grid of 12x12 neurons. The conclusions are the same. We present in Fig. 1.b, the evolution of the integrated $M_i(k)(\overline{M_1})$. The values are higher for DSOM and also less sensible to different values of γ. Figure 3 illustrates the central part of the map for $\gamma = 1$, where the organization of the referents with length of the words is evident. On this figure, only referent 117 ("present") does not belong to its partition. On the whole map, it is the case for 5 over 144 referents (3.5%). For DSOM(K) and DSOM(EG), the results are 23.4% and 99.7% respectively. From these characteristics, we also observe a higher effectiveness of the proposed algorithm which is mainly due to the implicit reference.

FUTURE TRENDS

The proposed algorithm is based on the computation of a "pseudo" gravity centre for each prototype. This computing is justified by assumption of existence of a latent Euclidean space. That means the dissimilarity data must be isometric to a L2 norm. In practice, this requirement is very seldom strictly checked and an approximation is often sufficient. Therefore, to completely validate this new DSOM, it is necessary to test it with more other data types and larger databases having a "ground truth". The data organization is interpreted after projection into the final map, and the neighbourhood in the output map must reveal the main latent properties of the observations which must be in agreement with the "ground truth".

N

CONCLUSION

This article presents a new affective algorithm for DSOM. Through the criteria of **trustworthiness** and **continuity**, this DSOM presents good topology preservation. The main reason of this improvement comes from the representation step where it is possible to continuously adapt the referent of each prototype like with the vector model. To achieve it, we use an implicit reference during the representation step thanks to the Huygens theorem. Even if the Euclidean assumptions are not exactly verified in practice, the distortions due to this mismatching are in fact less important than the ones occurred with the collision effect which is a difficult problem for the classical DSOM algorithms. This effectiveness is represented in this article by the better performance of the proposed algorithm compared to the other ones.

ACKNOWLEDGMENTS

REFERENCES

Ambroise C. and Govaert G. (1996). Analyzing dissimilarity matrices via Kohonen maps. IFCS-96, Int. Federation of Classification Societies, (2), Kobe (Japan), 96-99.

Barbara Spillmann. (2004). Description of the distance matrices. Institute of Computer Science and Applied Mathematics, University of Bern.

Borg I., Groenen P. (1997). Modern Multidimensional Scaling: Theory and Applications. Springer Verlag, New-York, Inc.

Conan-Guez B., Rossi F., El Golli A. (2006). Fast algorithm and implementation of dissimilarity self-organizing maps. Neural Networks, 19(6-7), 855-863.

El Golli A., Conan-Guez B., Rossi F. (2004). A self organizing map for dissimilarity data. IFCS-04, International Federation of Classification Societies, Chicago, 61-68.

Jain A.K., Dubes R.C. (1988). Algorithms for clustering Data, Prentice-Hall, Englewood Cliffs, NJ.

Joly S., Le Calvé G., (1994). Similarity functions, Chapter 3, 67-86, in Classification and Dissimilarity Analysis, Lecture Notes in Statistics, Van Cutsem ed., Springer-Verlag, New York.

Graepel T., Obermayer K. (1999). A stochastic self-organizing map for proximity data. Neural Computation, 11(1), 139–155.

Kohonen T. (1997). Self-Organizing Maps. Springer Verlag New York.

Kohonen T., Somervuo P.J. (1998). Self-organizing maps for symbol strings. Neurocomputing, (21), 19-30.

Kohonen T., Somervuo P.J. (2002). How to make large self-organizing maps for non vectorial data. Neural networks, 21(8).

Levenshtein V.I. (1966). Binary codes capable of correcting deletions, insertions and reversals, Soviet Physics Dokl., vol. 10 (8), 707-710.

Martínez C. D., Juan A., Casacuberta F. (2001). Improving classification using median string and NN rules. IX Spanish Symp. on Pattern Recog. and Image Analysis, (2), 391-395.

Van Cutsem B. (1994). Classification and Dissimilarity Analysis, Lecture Notes in Statistics, Van Cutsem Ed., Springer-Verlag, New York.

Venna J., Kaski S. (2001). Neighborhood preservation in nonlinear projection methods: An experimental study. ICANN 2001, Berlin, 485-491.

KEY TERMS

Affectation Step: A part of the learning iteration where an observation is affected to the nearest prototype according to a predefined distance.

Dissimilarity Data: Data in which all we know about the observations are pair-wise dissimilarities.

Dissimilarity SOM: A SOM where all observations are described by a dissimilarity matrix.

Prototype: Referent of a node (neuron) on the map.

Quantization Error: Error which appears when an observation is represented by a prototype.

Representation Step: A part of the learning iteration where the prototype is adapted to well represent its affected observations.

Self-Organizing Map (SOM): A subtype of artificial neural networks. It is trained using unsupervised learning to produce low dimensional representation of the training samples while preserving the topological properties of the input space.

SOM Batch Algorithm: A version of SOM in which at an iteration all observations are available and used for computation.

Topology Preservation: Preservation of the neighbourhood relation of the observations in the output space. It means that the observations which are neighbours in the input space should be projected in neighbour nodes.

NLP Techniques in Intelligent Tutoring Systems

Chutima Boonthum
Hampton University, USA

Irwin B. Levinstein
Old Dominion University, USA

Danielle S. McNamara
The University of Memphis, USA

Joseph P. Magliano
Northern Illinois University, USA

Keith K. Millis
The University of Memphis, USA

INTRODUCTION

Many Intelligent Tutoring Systems (ITSs) aim to help students become better readers. The computational challenges involved are (1) to assess the students' natural language inputs and (2) to provide appropriate feedback and guide students through the ITS curriculum. To overcome both challenges, the following non-structural Natural Language Processing (NLP) techniques have been explored and the first two are already in use: word-matching (WM), latent semantic analysis (LSA, Landauer, Foltz, & Laham, 1998), and topic models (TM, Steyvers & Griffiths, 2007).

This article describes these NLP techniques, the iSTART (Strategy Trainer for Active Reading and Thinking, McNamara, Levinstein, & Boonthum, 2004) intelligent tutor and the related Reading Strategies Assessment Tool (R-SAT, Magliano *et al.*, 2006), and how these NLP techniques can be used in assessing students' input in iSTART and R-SAT. This article also discusses other related NLP techniques which are used in other applications and may be of use in the assessment tools or intelligent tutoring systems.

BACKGROUND

Interpreting text is critical for intelligent tutoring systems (ITSs) that are designed to interact meaningfully with, and adapt to, the users' input. Different ITSs use different Natural Language Processing (NLP) techniques in their system. NLP systems may be structural, *i.e.,* focused on grammar and logic, or non-structural, *i.e.,* focused on words and statistics. This article deals with the latter.

Examples of the structural approach include ExtrAns (Extracting Answers from technical texts question-answering system; Molla *et al.*, 2003) which uses minimal logical forms (MLF; that is, the form of first order predicates) to represent both texts and questions and C-Rater (Leacock & Chodorow, 2003) which scores short-answer questions by analyzing the conceptual information of an answer in respect to the given question. Turning to the non-structural approach, AutoTutor (Graesser *et al.*, 2000) uses LSA to analyze the student's input against expected sets of answers and CIRCSIM-Tutor (Kim *et al.*, 1989) uses a word-matching technique to evaluate students' short answers. The systems considered more fully below, iSTART (McNamara *et al.*, 2004) and R-SAT (Magliano *et al.*, 2006) use both word-matching and LSA in assessing quality of students' self-explanation. Topic models (TM) were explored in both systems, but have not yet been integrated.

MAIN FOCUS OF THE CHAPTER

This article presents three non-structural NLP techniques (WM, LSA, and TM) which are currently used

or being explored in reading strategies assessment and training applications, particularly, iSTART and R-SAT.

Word Matching

Word matching is a simple and intuitive way to estimate the nature of an explanation. There are two ways to compare words from the reader's input (either answers or explanations) against benchmarks (collections of words that represent a unit of text or an ideal answer): (1) Literal word matching and (2) Soundex matching.

Literal word matching – Words are compared character by character and if there is a match of sufficient length then we call this a *literal match*. An alternative is to count words that have the same stem (*e.g.,* indexer and indexing) as matching. If a word is short a complete match may be required to reduce the number of false-positives.

Soundex matching – This algorithm compensates for misspellings by mapping similar characters to the same soundex symbol (Christian, 1998). Words are transformed to their soundex code by retaining the first character, dropping the vowels, and then converting other characters into soundex symbols: *1* for *b, p; 2* for *f, v; 3* for *c, k, s; etc.* Sometimes only one consecutive occurrence of the same symbol is retained. There are many variants of this algorithm designed to reduce the number of false positives (*e.g.,* Philips, 1990). As in literal matching, short words may require a full soundex match while for longer words the first *n* soundex symbols may suffice.

Word-matching is also used in other applications, such as, CIRCSIM-Tutor (Kim *et al.*, 1989) on short-answer questions and Short Essay Grading System (Ventura *et al.*, 2004) on questions with ideal expert answers.

Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA; Landauer, Foltz, & Laham, 1998) uses statistical computation to extract and represent the meaning of words. Meanings are represented in terms of their similarity to other words in a large corpus of documents. LSA begins by finding the frequency of terms used and the number of co-occurrences in each document throughout the corpus and then uses a powerful mathematical transformation to find deeper meanings and relations between words.

When measuring the similarity between text-objects, LSA's accuracy improves with the size of the objects, so it provides the most benefit in finding similarity between two documents but as it does not take word order into account, short documents may not receive the full benefit. The details for constructing an LSA corpus matrix are in Landauer & Dumais (1997). Briefly, the steps are: (1) select a corpus; (2) create a term-document-frequency (TDF) matrix; (3) apply Singular Value Decomposition (SVD; Press *et al.*, 1986) to the TDF matrix to decompose it into three matrices (L x S x R; where S is a scaling, matrix). The leftmost matrix (L) becomes the LSA matrix of that corpus. The optimal size is usually in the range of 300–400 dimensions. Hence, the LSA matrix dimensions become N x D where N is the number of unique words in the entire corpus and D is the optimal dimension (reduced from the total number of documents in the entire corpus).

The similarity of terms (or words) is computed by comparing two rows, each representing a term vector. This is done by taking the cosine of the two term vectors. To find the similarity of sentences or documents, (1) for each document, create a document vector using the sum of the term vectors of all the terms appearing in the document and (2) calculate a cosine between two document vectors. Cosine values range from ±1 where +1 means highly similar.

To use LSA in the tutoring systems, a set of benchmarks are created and compared with the trainee's input. Examples benchmarks are the current target sentence, previous sentences, and the ideal answer. A high cosine value between the current sentence benchmark and the reader's input would indicate that the reader understood the sentence and was able to paraphrase what was read. To provide appropriate feedback, a number of cosines are computed (one for each benchmark). Various statistical methods, such as discriminant analysis and regression analysis, are used to construct the feedback formula. McNamara *et al.* (2007) describe various ways that LSA can be used to evaluate the reader's explanations: either LSA alone or a combination of LSA with WM. The final conclusion is that a fully-automated (*i.e.*, less hand-crafted benchmarks construction), combined system produces the better results.

There are a number of other intelligent tutoring systems that use LSA in their feedback system, for examples, Summary Street (Steinhart, 2001), Auto-

Tutor (Greasser *et al.*, 2000), and Tutoring System (Lemaire, 1999).

Topic Models

The Topic Models approach (TM; Steyvers & Griffiths, 2007) applies a probabilistic model to find a relationship between terms and documents in terms of topics. A document is considered to be generated probabilistically from a number of topics where each topic consists of a number of terms, each given a probability of selection if that topic is used. By using a TM matrix, the probability that a certain topic was used in the creation of a given document is estimated. If two documents are similar, the estimates of the topics within these documents should be similar. TM is similar to LSA, except that a term-document frequency matrix is factored into two matrices instead of three: one is the probabilities of terms belonging to the topics (the TM matrix), the other the probabilities of topics belonging to the documents. The Topic Modeling Toolbox (Steyvers & Griffiths, 2007) can be used to construct a TM matrix,

To measure the similarity between documents, the Kullback Leibler distance (KL-distance: Steyvers & Griffiths, 2007) is recommended, rather than the cosine measure (which can also be used). Using TM in a tutoring system is similar to using LSA, where a set of benchmarks is defined and the reader's input is compared against each benchmark. The only different is the use of KL-distance instead of LSA-cosine value. The preliminary results of investigating TM in place of LSA (Boonthum, Levinstein, & McNamara, 2006) indicate that TM is as good as LSA alone (correlation between computerized-scores and human rating scores), but a little bit lower than a combined system using both WM and LSA. This suggests that the TM should be further investigated in combination with WM or LSA or both.

TM is mostly used in document clustering (grouping documents based on relevancy or similar topics; Buntine *et al.*, 2005), data mining (Tuulos & Tirri, 2004), and search engines (Perkiö *et al.*, 2004). A variation on TM by Steyvers & Griffiths (2007), is *Probabilistic Latent Semantic Analysis* (PLSA; Hofmann, 2001) which models each document as generated from a number of hidden topics and each topic has its features defined as the conditional probabilities of word occurrences in that topic.

iSTART and RSAT Applications

iSTART (Interactive Strategy Trainer for Active Reading and Thinking) is a web-based, automated tutor designed to help students become better readers using multi-media technology. It provides adolescent to college-aged students with a program of self-explanation and reading strategy training (McNamara *et al.*, 2004) called Self-Explanation Reading Training, or SERT (see McNamara *et al.*, 2004). iSTART consists of three modules: Introduction (description of SERT and reading strategies), Demonstration (illustration of how these reading strategies can be used), and Practice (hands-on practice of these reading strategies). In the Practice module, students practice using reading strategies by typing self-explanations of sentences. The system evaluates each explanation and then provides appropriate feedback to the student. If the explanation is irrelevant or too short compared to the given sentence and passage, the student is required to add more information. Otherwise, the feedback is based on the level of its overall quality.

The computational challenge is to provide appropriate feedback to the students about their explanations. Doing so requires capturing some sense of both the meaning and quality of their explanation. A combination of word-matching and LSA provided better results (comparing the computerized-score using NLP techniques to the human rating score and having higher correlation between these two sets of scores) than either separately (McNamara, Boonthum, Levinstein, & Millis, 2007).

R-SAT (Reading Strategy Assessment Tool; Maglino *et al.*, 2007) is an automated web-based reading assessment tool designed to measure readers' comprehension and spontaneous use of reading strategies. The R-SAT is similar to the iSTART Practice module in the sense that it presents passages to the reader one sentence at a time and asks for the reader's input. The difference is that, instead of an explanation, R-SAT asks either an indirect ("What are your thoughts regarding your understanding of the sentence in the context of the passage?") or a direct question (*e.g.,* Why did the miller want to marry the girl?") at pre-selected target sentences. The answers to the indirect questions are evaluated on how they are related to the given sentence and passage; the answers to the direct questions are assessed by comparing them to ideal answers.

The problem is to analyze the answers and generate a set of scores for overall comprehension and strategy usage. Ultimately, these scores can be used as a pre-assessment for iSTART allowing the trainer to individualize the iSTART curriculum based on the reader's needs. R-SAT was initially proposed to use word-matching, LSA, and other techniques beyond LSA. However, during the course of development, word-matching was found to produce better results than LSA or in combination with LSA.

FUTURE TRENDS

These three NLP techniques (WM, LSA, and TM) are used in the ongoing research on assessing and improving comprehension skills via reading strategies in the R-SAT and iSTART projects. WM and LSA have been extensively investigated for iSTART and to some extent in R-SAT. The lack of success of LSA compared to the simpler WM in R-SAT is somewhat surprising and may be due to particular features of the algorithms used or to the variety of text genres used in R-SAT. Future work is planned with modified algorithms and substituting genre-specific LSA spaces for the general space now used. In addition TM needs further exploration, especially in its use with small units of text where the recommended Kullback Leibler distance has not proven particularly effective.

CONCLUSION

The purpose of this article is to describe three NLP techniques and how they can be used in assessment tools and intelligent tutoring systems. For iSTART to teach reading strategies effectively, it must be able to deliver valid feedback on the quality of the explanations that a reader produces and therefore the system must understand, at least to some extent, the explanation. Of course, automating natural language understanding has been extremely challenging, especially for non-restrictive content domains like explaining a freely-entered text t. Algorithms such as LSA open up a number of possibilities to systems such as iSTART: in essence LSA provides a 'simple' algorithm that allowed tutoring systems to provide appropriate feedback to students (see Landauer *et al.*, 2007). The results presented in Boonthum *et al.* (2006) show that the topic model similarly offers a wealth of possibilities in natural language processing. For R-SAT to measure a reader's comprehension and reading skills accurately, like iSTART it must also be able to understand, to some extent, what a reader says, especially when he/she is asked to describe their current thoughts. Although LSA is a good candidate, simple word matching against various benchmarks seems adequate to provide satisfactory results especially when aggregated over several explanations (see Magliano *et al.*, 2006). It is also demonstrates that a combination of techniques produces better results than using one technique on its own.

REFERENCES

Boonthum, C., Levinstein, I.B., & McNamara, D.S. (2006). Evaluating Self-Explanations in iSTART: Word Matching, Latent Semantic Analysis, and Topic Models. In A. Kao & S. Poteet (Eds.), *Text Mining and Natural Language Processing*, Springer. 91-106.

Buntine, W., Löfström, J., Perttu, S., & Valtonen, K. (2005). Topic-Specific Scoring of Documents for Relevant Retrieval. In *Workshop on Learning in Web Search (LWS-2005)*, pp 34-41.

Christian. P. (1998). Soundex – can it be improved? *Computers in Genealogy*, 6 (5).

Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., & TRG. (2000). Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, 8 , 149-169.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, pp 177-196.

Kim, N., Evens, M.W., Michael, J.A., & Rovick, A.A. (1989). CIRCSIM-Tutor: An Intelligent Tutoring System for Circulatory Physiology. In Maurer, H. (ed.), *Computer-Assisted Learning: 2nd International Conference* (ICCAL-89), pp. 254-266. Berlin: Springer-Verlag.

Landauer, T.K., Foltz, P.W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259-284.

Landauer, T., McNamara, D.S., Dennis, S., & Kintsch, W. (2007). *A Handbook of Latent Semantic Analysis* Mahwah, NJ: Erlbaum.

Landauer, T.K. & Dumais, S.T. (1997) A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. Psychological Review, 104(2), 211-240.

Leacock, C., & Chodorow, M. (2003). C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4), 389-405.

Lemaire, B. (1999). Tutoring Systems Based on Latent Semantic Analysis. In *Artificial Intelligence in Education* (AIED-99), S. Lajoie and M. Vivet (eds.), IOS Press, Amsterdam, pp. 527-534.

Magliano, J.P., Millis, K.K., Gilliam, S., Levinstein, I.B., & Boonthum, C. (2006). Assessing Reading Comprehension with Verbal Protocols and Latent Semantic Analysis. In *the Proceeding of the 47th Annual Meeting of the Psychonomic Society*, Houston, TX.

McNamara, D.S., Boonthum, C., Levinstein, I.B., & Millis, K.K. (2007). Using LSA and word-based measures to assess self-explanations in iSTART. In T. Landauer *et al.* (Eds.), *A Handbook of Latent Semantic Analysis* (pp. 227-241). Mahwah, NJ: Erlbaum.

McNamara, D.S., Levinstein, I.B., & Boonthum, C. (2004). iSTART: Interactive Strategy Trainer for Active Reading and Thinking. Submitted to *Behavioral Research Methods, Instruments, and Computers*, 36, 222-233.

Molla, D., Schwitter, R., Rinaldi, F., Dowdall, J., & Hess, M. (2003). ExtrAns: Extracting Answers from Technical Texts. *IEEE Intelligent System*, 18(4): 12-17.

Perkiö, J., Buntine., W., & Perttu, S. (2004). Exploring Independent Trends in a Topic-Based Search Engine. In *Proceedings of the Web Intelligence Conference (WI-2004)*, pp. 664-668.

Philips, L. (1990). Hanging on the Metaphone. *Computer Language,* 7(12).

Press, W.M., Flannery, B.P., Teukolsky, S.A., & Vetterling, W.T. (1986). Numerical recipes: The art of scientific computing. New York, NY: Cambridge University Press.

Steinhart, D. (2001). Summary Street: An intelligent tutoring system for improving student writing through the use of latent semantic analysis. *Ph.D. dissertation*, Dept. Psychology, Univ. Colorado, Boulder.

Steyvers, M., & Griffiths, T. (2007). Probabilistic Topic Models. In T. Landauer, D.S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of Latent Semantic Analysis* (pp. 427-448). Mahwah, NJ: Erlbaum.

Tuulos, V.H. & Tirri, H. (2004). Combining Topic Models and Social networks for Chat Data Mining. In *Proceedings of on Web Intelligence Conference (WI-2004)*, pp. 206-213.

Ventura, M.J., Franchescetti, D.R., Pennumatsa, P., Graesser, A.C., Jackson, G.T., Hu, X., Cai, Z., & TRG. (2004). Combining Computational Models of Short Essay Grading for Conceptual Physics Problems. In J.C. Lester *et al.* (Eds.), *Intelligent Tutoring Systems* (pp. 423-431). Berlin, Germany: Springer.

KEY TERMS

Intelligent Tutoring System (ITS): Also called Intelligence Computer-Aided Instruction (ICAI), a personal training assistant that captures the subject matter and teaching expertise and individualize the curriculum to meet each learner's needs in order to master the subject matter. Its main goal is to provide benefits of the one-on-one instruction: lessons are conducted at the learner's own pace; practices are interactive so the learner can improve their weaker skills; and real-time question answering clarify learner's doubts or misunderstanding; and an individualized curriculum based on the learner's needs.

Kullback Leibler Distance (KL-distance): A natural distance function from a "true" probability distribution to a "target" probability distribution. It can be interpreted as the expected extra message-length per datum due to using a code based on the wrong (target) distribution compared to using a code based on the true distribution.

Latent Semantic Analysis (LSA): A natural language processing technique that analyses relationships between a set of documents and terms within these documents. LSA was created in 1990 for informa-

tion retrieval and is sometimes called latent semantic indexing (LSI).

LSA Cosine: A measurement of a relation between two vector-units. A unit can be as small as a word or as large as an entire document. It can be computed using the dot-product of two vectors where each vector is a representation of a unit (word, sentence, paragraph, or whole document).

Probabilistic Latent Semantic Analysis (PLSA): A statistical techniques for the analysis of two-mode and co-occurrence data, which has applications in information retrieval and filtering, natural language processing, machine learning from text, and related areas. PLSA evolved from LSA but focuses more on the relationship of topics within documents.

Protocols: Any verbal input that students or readers produce during a session. This can be a set of explanations or answers to direct questions.

Self-Explanation and Reading Strategy Trainer (SERT): Pedagogy uses five strategies to help students become a better reader. The reading strategies include (1) comprehension monitoring, being aware of one's own understanding of the text; (2) paraphrasing, or restating the text in different words; (3) elaboration, using prior knowledge or experiences to understand the text (domain-specific knowledge-based inferences) or using common-sense or logic to understand the text (general knowledge based inferences); (4) predictions, predicting what the text will say next; and (5) bridging, understanding the relation between separate sentences of the text.

Word Matching (WM): A simple way to compare words. Literal match is done by comparing character by character, while Soundex match transforms each word into a Soundex code, similar to phonetic spelling.

Non–Cooperative Facial Biometric Identification Systems

Carlos M. Travieso González
University of Las Palmas de Gran Canaria, Spain

Aythami Morales Moreno
University of Las Palmas de Gran Canaria, Spain

INTRODUCTION

The verification of identity is becoming a crucial factor in our hugely interconnected society. Questions such as "Is she really who she claims to be?", "Is this person authorized to use this facility?" are routinely being posed in a variety of scenarios ranging from issuing a driver's license to gaining entry into a country. The necessity for reliable user authentication techniques has increased in the wake of heightened concerns about security and rapid advancements in networking, communication, and mobility. Biometric systems, described as the science in order to recognize an individual based on his or her physical or behavioural traits, is beginning to get acceptance as a legitimate method in order to determine an individual's identity. Nowadays, biometric systems have been deployed in various commercial, civilian, and forensic applications as a means of establishing identity.

In particular, this work presents a non-cooperative identification system based on facial biometric.

BACKGROUND

How do biological measurements qualify as being biometric? Any human physiological and/or behavioural characteristic can be used as a biometric characteristic as long as it satisfies the following requirements (Jain, Ross & Prabhakar, 2004): universality, distinctiveness, permanence, collectability.

The choice of biometric identifiers has a major impact on the performance of the system. This choice depends greatly on the intended application of the system. Currently, some of the most widely used biometrics identifiers include fingerprints (Jain, Ross & Prabhakar, 2004, pp. 43-64), hand geometry (Sanchez-Reillo, Sanchez-Avila, Gonzalez-Marcos, 2000), iris (Jain, Ross & Prabhakar, 2004, pp. 103-121), face (Jain, Ross & Prabhakar, 2004, pp. 65-86), etc...

Most biometric systems require co-operation on the part of the users in order to acquire their biometric data. Face identification, however, does not require this condition for its use, although it can be used. This is therefore its principal advantage over other biometric systems. Human face identification is an extensively studied field since the computational cost has not been turned out to be a drawback, due to the increasing importance of this kind of biometric identification in the access security to places such as airports, metros, train and bus stations. The process of facial identification incorporates two significant methods: detection (an individual from among a set) and identification (whether an individual is whom s/he claims to be).

Face detection (Young-Bum Sun, Jin-Tae Kim & Won-Hyung Lee, 2002) involves locating the human face within an image captured by a video camera and taking that face and isolating it from the other objects captured within the image.

Identification is comparing the captured face with other faces that have been saved and stored in a database. The basic underlying identification technology of facial feature identification involves either eigenfeatures (facial metrics) or eigenfaces. Within this type of study a great variety of references can be found (Discrete Cosine Transform (DCT), Karhunen-Loeve (KL) Transform, Independent Component Analysis (ICA), Principal Component Analysis (PCA), etc). The greatest advantage of a facial identification system is its non-cooperative nature as it is a system which can work independently of user co-operation.

FACIAL IDENTIFICATION SYSTEM

This article presents the two principal processes associated with face identification: face detection and face identification. However, there also exist other aspects of facial identification system to be taken into account. In the face detection module the face capturing is shown, just when the camera takes a picture or frame. The image acquisition can be carried out using RGB images, Infrared (IR) images among other formats; recently thermal images are also being used. The choice of the image format depends on its applications, lighting conditions, location (indoor or outdoor system), and the degree of security.

In the face identification module, a database can be found with the user information that must be located; therefore a supervised classification must be carried out. The parametrization submodule extracts the user features, and the classification system generates a model in order to difference our user/users versus the remainder of persons (see figure 1).

Face Detection

The challenges associated with face detection can be attributed to the following factors: Pose, presence or absence of structural components, facial expression, occlusion, image orientation, imaging conditions.

There are many closely related problems with respect to face detection. Face localization aims to determine the image position of a single face; this is a simplified detection problem with the assumption that an input image contains only one face (Lam & Yan, 1994). The goal of facial feature detection is the detection of the presence and location of features, such as eyes, nose, nostrils, eyebrow, mouth, lips, ears, etc., with the assumption that there is only one face in an image (Zhiwei, & Oiang, 2006). Face recognition or face identification compares an input image against a database and reports a match, if found (Darrell, Gordon, Harville & Woodfill, 2000). The purpose of face authentication is to verify the claim of the individual's identity in an input image (Crowley & Berard, 1997), while face tracking methods continuously estimate the location and possibly the orientation of a face in an image sequence in real time (Darrell, Gordon, Harville, & Woodfill, 2000, Zhiwei, & Qiang, 2006) (see figure 2).

Several face detection systems have been introduced (Ming-Hsuan Yang, David Kriegman & Narendra Ahuja, 2002) (Yang, Ahuja, &Kriegman, 2000). There are many existing techniques to detect faces based on a single image. The techniques for face detection with a single image were classified into three categories.

- **Knowledge Based System:** This approach depends on using rules about human facial features to detect faces. Human facial features such as two eyes that are symmetric to each other, a nose and mouth, and other distance features represent this feature set. After detecting features, a verification process is carried out to reduce false detection. This approach is good for frontal images, as is shown in figure 3. The difficulty lies in translating human knowledge into known rules and to detect faces in different poses.
  Furthermore, the surrounding environment can also pose a problem. For example, changes in light sources can add or remove shadows from a face. Therefore, many variables should be considered when designing a face detection system.

Figure 1. Block diagram for a non-cooperative facial identification
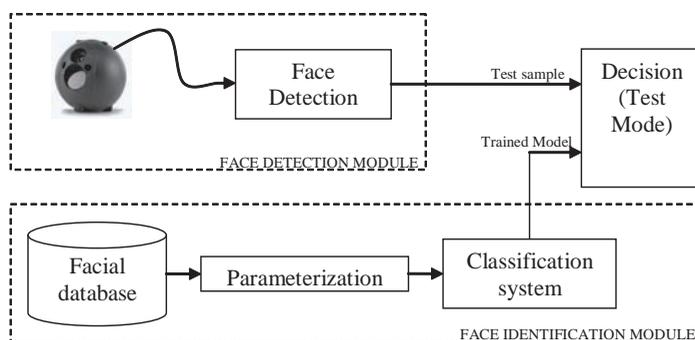
Figure 2. Face detection examples in a motion picture captures
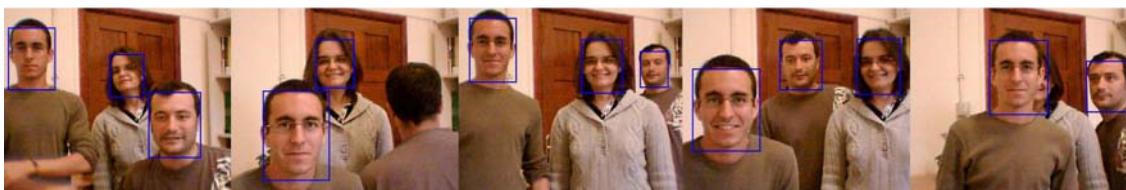
For these reasons, in a non-cooperative system this technique suffers invariability.

- **Image Based System:** In this approach, a predefined standard face pattern is used to match with the segments in the image to determine whether they are faces or not. It uses training algorithms to classify regions into face or non-face classes. Image-based techniques depend on multi-resolution window scanning to detect faces, so these techniques have high detection rates but are slower than the feature-based techniques. Eigenfaces (Yang, Ahuja, & Kriegman, 2000) and neural networks (Rowley, Baluja & Kanade, 1998) are examples of image-based techniques. This approach has the advantage of being simple to implement, but it cannot effectively deal with variation in scale, pose and shape (Rein-Lien Hsu & Jain, 2002).

- **Features Based System:** This approach depends on extraction of facial features, which are not affected by variations of lighting conditions, pose, and/or other factors. These methods are classified

Figure 3. A typical face image used in knowledge based methods

according to the extracted features. Feature-based techniques depend on feature derivation and analysis to gain the required knowledge about faces. Features may be skin colour, face shape, or facial features such as eyes, nose, etc.... Feature based methods are preferred for real time systems where the multi-resolution window scanning used by image based methods are not applicable. Human skin colour is an effective feature used to detect faces, because although different people have different skin colours, several studies have shown that the basic difference is based on their intensity rather than their chrominance. Human faces have a special texture that can be used to separate them from different objects (Bojkovic, & Samcovic, 2006). The facial features method depends on detecting features of the face.

Face Identification in Transform Domain Systems

The detected faces always have variable conditions (lighting, expression, rotation, translation, etc), and therefore, images used to train can have some changes with respect to images from face detection. The use of Features or Knowledge Based Systems is a disadvantage due to the wide data variability from variable conditions. Therefore, transform domain systems are a good goal because they group the information and contribute more discrimination to the facial identification.

Transform domain analysis is a commonly used image processing and a parameterization technique. In recent years some work has been done to extract transform domain features for image identification. Li et al. extract Fourier range and angle features to identify the palm-print image (Li, Zhang & Xu, 2002). Lai et al. use holistic Fourier invariant features to recognize the facial image (Lai, Yuen & Feng, 2001). Another spectral

feature generated from singular value decomposition (SVD) is used by some researchers (Chellappa, Wilson & Sirohey, 1995). However, Tian et al. indicate that this feature does not contain adequate information for face recognition (Tian, Tan, Wang & Fang, 2003). Hafed and Levine (2001) extract discrete cosine transform (DCT) feature for face recognition. They point out that DCT obtains the near-optimal performance of Karhunen–Loeve (KL) transform in facial information compression. And the performance of DCT is superior to those of discrete Fourier transform (FT) and other conventional transforms. By manually selecting the DCT frequency bands, their recognition method achieves a similar recognition effect to the Eigenface method (M. H. Yang, 2002) which is based on KL transform. Nevertheless, their method cannot provide a rational band selection rule or strategy. Nor can it outperform the classic Eigenface method.

In addition, some extended discrimination methods are proposed. Zhang *et al.* (2002) present a dual Eigenspace method for face recognition. In his work, W. Malina (2001), proposed several new discrimination principles based on the Fisher criterion. Yang uses principal component analysis kernel (PCA) for facial feature extraction and recognition (Bartlett, Movellan & Sejnowski,, 2002), while Bartlett *et al.* (2002) apply the independent component analysis (ICA) in face recognition. However, Yang shows that both ICA and PCA kernels need much more computing time than PCA. In addition, when the Euclidean distance is used, there is no significant difference in the classification performance of PCA and ICA (Bartlett, Movellan & Sejnowski, 2002). Jing *et al.* (2003) put forward a classifier combination method for face recognition. This paper does not analyze and compare these extended

discrimination methods, but limits itself to a comparison of major linear discrimination methods including the Eigenface method, the Fisherface method, DLDA and discriminated waveletface.

The KL transform is an optimal transform for removing statistical correlation. Of the discrete transforms, DCT approaches the KL transform (Hu, Worrall, Sadka & Kondoz, 2001). In other words, DCT has strong ability to remove correlation and compress images. Furthermore, DCT can be used by fast Fourier transform (FFT), while there is no fast realization algorithm for KL transform. Therefore, our approach sufficiently uses these favourable properties of DCT.

The following table shows different systems based on different methods of face recognition with their corresponding recognition rates. The databases used are ORL [ORL Database], Yale [Yale Database], AR-Face [AR Database] and FERET [FERET Database].

FUTURE TRENDS

Recently, numerous methods that combine several facial features have been proposed to locate or detect faces. Most of them use global features such as skin colour, size, and shape to find face candidates, and then verify these candidates using different local parameterization methods. The challenge is to achieve invariability of the captured images from the conditions (light, shapes ...) and positional changes (rotations, scales …). The creation and development of new methods based on transform domain system will provide robust characteristics for achieving this invariability.

With respect to facial identification, 3D techniques can be used for the purpose in this system, but the

Figure 4. Face samples with different conditions (lighting and rotation)

computational cost is a major disadvantage for real time applications. Facial rebuilding with 3D techniques can obtain more information and any features can be extracted. Moreover, this system retains the non-co-operation quality. In the future, the use of the multi-modal systems with other biometric characteristics will generate a stronger and robust system.

CONCLUSION

Face recognition is a challenging and interesting problem. However, it can also be regarded as part of the wider attempt to solve one of the greatest challenges to computer vision, that of object recognition. In particular, facial identification is becoming a very important biometric system in the battle to reduce global terrorism. Much research has already been carried out in this field, and bearing in mind the threat to security which the world is currently facing, there will undoubtedly be many more publications on facial identification in the future.

REFERENCES

Jain, A. K., Ross, A., & Prabhakar, S., (2004), An introduction to Biometric Recognition, *IEEE Transactions on Circuits and Systems for Video Technology*, Special Issue on Image and Video Based Biometrics, 14(1), pp.4 - 20.

Sanchez-Reillo, R., Sanchez-Avila, C., & Gonzalez-Marcos, A., (2000), Biometric identification through hand geometry measurements, in *IEEE Transactions on Pattern Analisys and Machine Intelligence*, 22(10), pp. 1168-1171.

Young-Bum S., Jin-Tae K., & Won-Hyung L., (2002) Extraction of face objects using skin color information, in *IEEE 2002 International Conference* on Communications, 1, pp. 600 - 604.

Lam, K., & Yan, H., (1994) Fast Algorithm for Locating Head Boundaries, *J. Electronic Imaging*, 3(4), pp. 351-359.

Zhiwei, Z., & Qiang J., (2006) Robust Pose Invariant Facial Feature Detection and Tracking in Real-Time, *18th International Conference on Pattern Recognition* 1, pp. 1092 – 1095.

Table 1. Results of different systems with the different databases

Databases	Systems	Recognition Rates
ORL	IGF (Liu & Wechsler, 2003)	100%
	Gabor with FLD (Zhu, Vai & Peng Un Mak, 2004)	99,0%
	Discrete Wavelet Transform + SVM (Travieso et al., 2004)	98,9%
	FRCM (Ho-Man Tang, Michael Lyu & Irwin King, 2003)	98,8%
	ENFS (Zhu, Vai & Mak, 2004)	98,5%
	Embedded HMM (Nefian & Hayes, 1999)	98,0%
	Several SVM+NN arbitrator (Kim, Jung & Kim, 2002)	97,9%
	Kernel PCA (Kim, Jung & Kim, 2002)	97,5%
	Nearest Feature Space (Chien & Wu, 2002)	96,1%
	2D DCT with KPCA and NFS (Zhu, Vai & Mak, 2003)	96,0%
Yale	ICA + SVM (Déniz, Castrillón & Hernández, 2003)	99.3%
	Discriminative Common Vector (Cevikalp et al., 2005)	97.3%
	MRF (Huang, Pavlovic & Metaxas, 2004)	96,1%
	FRCM (Ho-Man Tang, Michael Lyu & Irwin King, 2003)	96,0%
AR-Face	Discriminative Common Vector (Cevikalp et al., 2005)	99.3%
FERET	Gabor + ICA (Liu & Wechsler, 2003)	100%
	ICA + SVM (Jain & Huang, 2004)	95.7%

Yang, M.H., Ahuja, N., & Kriegman, D., (2000), **Face recognition using kernel eigenfaces,** International Conference on Image Processing, 1, pp. 37 – 40.

Crowley, J. L., and Berard, F., (1997) Multi-Modal Tracking of Faces for Video Communications, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 640-645.

Darrell, T., Gordon, G., Harville, M., & Woodfill, J., (2000) Integrated Person Tracking Using Stereo, Color, and Pattern Detection, *Int'l J. Computer Vision*, vol. 37, no. 2, pp. 175-185.

Ming-Hsuan Yang, David J. Kriegman, & Narendra Ahuja, (2002) Detecting Faces in Images *IEEE Trans. Pattern Analysis And Machine Intelligence* , vol. 24, no. 1.

Rowley, H. A., Baluja, S., & Kanade, T., (1998) Neural Network Based Face Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, January, pp. 23-38.

Bojkovic, Z., & Samcovic, A., (2006), Face Detection Approach in Neural Network Based Method for Video Surveillance, *8th Seminar on Neural Network Applications in Electrical Engineering,* pp. 44 – 47.

Li, W., Zhang, D., & Xu, Z., (2002) Palmprint identification by Fourier transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4), pp. 417–432.

Lai, J. H., Yuen, P. C., & Feng, G. C., (2001) Face recognition using holistic Fourier invariant features, *Pattern Recognition*, 34(1), pp. 95–109, 2001.

Chellappa, R., Wilson, C., & Sirohey, S., (1995) Human and machine recognition of faces: A survey, *Proceedings of IEEE*, 83, pp. 705–740.

Tian, Y., Tan, T. N., Wang, Y. H. & Fang, Y. C., (2003) Do singular values contain adequate information for face recognition?, *Pattern Recognition*, 36(3) pp. 649–655.

Hafed, Z. M., & Levine, M. D., (2001) Face recognition using the discrete cosine transform, *International Journal Computation Vision*, 43(3) pp. 167–188.

Zhang, D., Peng, H., Zhou, J., & Pal, S. K., (2002) A novel face recognition system using hybrid neural and dual eighefaces methods, *IEEE Transaction on System., Man, and Cybernetic. A*, 32, pp. 787–793.

Malina, W., (2001) Two-parameter Fisher criterion, *IEEE Transaction on System., Man, and Cybernetic B*, 31, pp. 629–636.

Yang, M. H., (2002) Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods, in *IEEE Proc. 5th International Conference Automatic Face Gesture Recognition*, pp. 215–220.

Bartlett, M. S., Movellan, J. R., & Sejnowski, T. J., (2002) Face recognition by independent component analysis, *IEEE Transaction on. Neural Network*, 13, pp. 1450–1464.

Jain, A. K., (1989) *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice–Hall.

Jing, X. Y., Zhang, D., & Yang, J. Y., (2003) Face recognition based on a group decision-making combination approach, *Pattern Recognition*, 36(7), pp. 1675–1678.

Liu, C., & Wechsler, H., (2003) *Independent component analysis of Gabor features for face recognition,* IEEE Transactions on Neural Networks, 14, pp. 919-928.

Zhu, J., Vai, M., & Peng U.M., (2004) Gabor Wavelets Transform and Extended Nearest Feature Space Classifier for Face Recognition, *Third International Conference on Image and Graphics,* pp. 246-249.

Tang, HM., Lyu, M., & King, I., (2003) Face recognition committee machine*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 837- 840.

Zhu, J., Vai, M. & Mak, P., (2004) A New Enhanced Nearest Feature Space (ENFS) Classifier for Gabor Wavelets Features-based Face Recognition*, International Conference on Biometric Authentication*, pp. 124-131.

Nefian, A.V., & Hayes, M.H., (1999) An embedded HMM-based approach for face detection and recognition, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 6, pp 3553-3556.

Kim, K. I., Jung, K. & Kim, H. J. , (2002) Face Recognition Using Kernel Principal Component Analysis, IEEE Signal Processing Letters, 9, pp. 40- 42.

Chien, J.T., & Wu, C.C., (2002) Discriminant wavelet-faces and nearest feature classifiers for face recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, pp. 1644-1649.

Zhu, J., Vai, M., & Mak, P., (2003) Face Recognition, a Kernel PCA Approach, *Chinese Conference on Medicine and Biology*, pp. 81-83.

Huang R., Pavlovic, V., & Metaxas, D., (2004) A hybrid face recognition method using Markov random fields, *Proceedings of the 17th International Conference on Pattern Recognition*, 3, pp. 157-160.

Travieso C.M., Alonso J.B., & Ferrer M.A., (2004) Facial identification using transformed domain by SVM, *Proceedings of the 38th IEEE International Carnahan Conference on Security Technology*, pp. 193-196.

Déniz O., Castrillón M., & Hernandez M., (2003) Face recognition using independent component analysis and support vector machines, *Pattern Recognition Letters*, 24(13), pp. 2153-2157.

Cevikalp H., Neamtu M., Wilkes M., & Barkana A., (2005) Discriminative Common Vectors fo*r Face Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1), pp. 4-13.

Jain A., & Huang J., (2004) Integrating Independent Components and Support Vector Machines for Gender Classification, *Proceedings of the 17th International Conference on Pattern Recognition*, 3, pp. 558-561.

ORL Database, http://www.uk.research.att.com/face-database.html (last visit: 07-31-05)

Yale Database, http://cvc.yale.edu/projects/yalefaces/yalefaces.html (last visit: 07-31-07)

AR-Face Database, http://cobweb.ecn.purdue.edu/~aleix/aleix_face_DB.html (last visit: 07-31-07)

FERET Database, http://www.itl.nist.gov/iad/humanid/feret/feret_master.html (last visit: 07-31-07)

KEY TERMS

N

Biometric System: This is a system which identifies persons from physical or behavioral characteristics. These characteristics are intrinsic to the individuals.

Face Detection: The act of detecting a face from a frame or an image.

Face Identification: This is a system which creates a model from facial features in order to recognize persons.

Independent Component Analysis (ICA): A computational method for separating a multivariate signal into additive subcomponents supposing the mutual statistical independence of the non-Gaussian source signals.

Multi-Modal System: Use of different biometric system in order to identify or verify persons.

Non-Cooperative Identification System: This is a system for identification which does not require the collaboration of a user in order to operate. The information for identification is obtained with the permission of the user.

Supervised Classification: Classification system that generates a model using training samples, and it uses that model to establish an evaluation or test with other samples.

Transform Domain System: This is a change from visible range to another different range, which transforms the information, providing other properties in this domain.

Nonlinear Techniques for Signals Characterization

Jesús Bernardino Alonso Hernández
University of Las Palmas de Gran Canaria, Spain

Patricia Henríquez Rodríguez
University of Las Palmas de Gran Canaria, Spain

INTRODUCTION

The field of nonlinear **signal characterization** and nonlinear signal processing has attracted a growing number of researchers in the past three decades. This comes from the fact that linear techniques have some limitations in certain areas of signal processing. Numerous nonlinear techniques have been introduced to complement the classical linear methods and as an alternative when the assumption of linearity is inappropriate. Two of these techniques are higher order statistics (HOS) and nonlinear dynamics theory (chaos). They have been widely applied to time series characterization and analysis in several fields, especially in biomedical signals.

Both HOS and chaos techniques have had a similar evolution. They were first studied around 1900: the method of moments (related to HOS) was developed by Pearson and in 1890 Henri Poincaré found sensitive dependence on initial conditions (a symptom of chaos) in a particular case of the three-body problem. Both approaches were replaced by linear techniques until around 1960, when Lorenz rediscovered by coincidence a chaotic system while he was studying the behaviour of air masses. Meanwhile, a group of statisticians at the University of California began to explore the use of HOS techniques again.

However, these techniques were ignored until 1980 when Mendel (Mendel, 1991) developed system identification techniques based on HOS and Ruelle (Ruelle, 1979), Packard (Packard, 1980), Takens (Takens, 1981) and Casdagli (Casdagli, 1989) set the methods to model nonlinear time series through chaos theory. But it is only recently that the application of HOS and chaos in time series has been feasible thanks to higher computation capacity of computers and Digital Signal Processing (DSP) technology.

The present article presents the state of the art of two nonlinear techniques applied to time series analysis: higher order statistics and chaos theory. Some measurements based on HOS and chaos techniques will be described and the way in which these measurements characterize different behaviours of a signal will be analized. The application of nonlinear measurements permits more realistic characterization of signals and therefore it is an advance in automatic systems development.

BACKGROUND

In digital signal processing, **estimators** are used in order to characterize signals and systems. These estimators are usually obtained using linear techniques. Their mathematical simplicity and the existence of a unifying linear systems theory made their computation easy. Furthermore, linear processing techniques offer satisfactory performance for a variety of applications.

However, linear models and techniques cannot solve issues such as nonlinearities due to noise, to the production system of the signal, system nonlinearities in digital signal acquisition, transmission and perception, nonlinearities introduced by the processing method and nonlinear dynamics behaviour. Therefore, the application of linear processing techniques leads to less realistic characterization of certain systems and signals. As a result of the shortcomings of linear techniques, analysis procedures are being revised and nonlinear techniques are being applied in computing **estimators** and models and in **signal characterization** to increase the possibilities of digital signal processing.

HOS is a field of statistical signal processing which has become very popular in the last 25 years. To date almost all digital signal processing have been based

on second order statistics (autocorrelation function, power spectrum). HOS use extra information which can be used to get better estimates of noisy situation and nonlinearities.

Chaos theory (nonlinear dynamical theory) is a long-term unpredictable behaviour in a nonlinear dynamic system caused by sensitive on initial conditions. Therefore, irregularities in a signal can be produced not only by random external input but also by chaotic behaviour.

Both nonlinear techniques have been used in signals characterization and numerous automatic classification systems have been developed using HOS and chaos features in many fields. Texture classification (Coroyer, Declercq, Duvaut, 1997), seismic event prediction (Van Zyl, 2001), fault diagnosis in machine condition monitoring through vibration signals (Samanta, Al.Balushi. & Al-Araimi, 2006), (Wang & Lin, 2003) and economy (Hommes & Manzan, 2006) are some examples.

Their application in biomedical signals is especially important. Nonlinear features have proven to be useful in voice, electrocardiogram (ECG) and electroencephalogram (EEG) signals characterization. Automatic classification systems between pathological and healthy voices have been implemented using nonlinear features (Alonso, de León, Alonso, Ferrer, 2001) (Alonso, Díaz-de-María, Travieso, Ferrer, 2005). Nonlinear characteristics have been used in the detection of electrocardiographic changes through ECG signal (Ubeyli & Guler, 2004), in the evaluation of neurological diseases using EEG signal (Gulera, Ubeylib & Guler, 2005), (Kannathal, Lim Choo Min, Rajendra Acharya & Sadasivan, 2005) and in diagnosis of phonocardiogram (Shen, Shen, 1997).

NONLINEAR METHODS: CHAOS THEORY AND HIGHER ORDER STATISTICS APPLIED TO TIME SERIES

Higher Order Statistics

Higher Order Statistics, known as cumulants and their Fourier transform, known as polyspectra are extensions of second-order measures (such as the autocorrelation function and power spectrum). Some advantages of HOS over second-order statistics are:

1. HOS give amplitude and phase information in the spectral domain, whereas second order statistics only give amplitude information (Mendel, 1991) (Nikias & Petropulu, 1993). Therefore, non-minimum phase signals and certain types of phase coupling (associated with nonlinearities) cannot be correctly identified by second-order statistics.
2. HOS are blind to Gaussian processes whereas correlation is not (Mendel, 1991). Therefore, cumulants can be used in determining Gaussian noise levels in a signal, separating non-Gaussian signals from Gaussian noise, in harmonics components estimation or in increasing signal to noise ratio (SNR) when signals are contaminated with Gaussian noise.

The second-order measures work properly if the signal has a Gaussian probability density function, but many real-life signals are non-Gaussian. Therefore, HOS are a powerful tool to work with non-Gaussian and nonlinear processes.

Next, some higher order statistics measurements are shown and their usefulness in characterizing certain nonlinear phenomena is explained.

Third Order Moment: Skewness

Skewness is a third order moment and a measure of the asymmetry in a probability distribution. This measurement enables us to discriminate among different kind of data distribution as its value varies according to the asymmetry of a distribution. The skewness of a Normal distribution is zero (data symmetric about the mean), positive skewness corresponds to a distribution with a right tail longer and negative skewness to a distribution with a left tail longer.

In most cases normal distribution is assumed, but data points are not usually perfectly symmetric. Skewness reflects positive or negative deviations from the mean and gives more realistic characterization of a data set.

Fourth Order Moment: Kurtosis

Kurtosis is a fourth order moment and a measure of whether the data in a probability distribution are peaked

or flat relative to a Normal distribution. Kurtosis is a measure of the data concentration about the mean, higher kurtosis means more of the variance is due to infrequent extreme deviations.

Higher Order Cumulants

Higher order moments are natural generalization of autocorrelation, while cumulant (Mendel, 1991) are nonlinear combinations of moments. The second order cumulant is the autocorrelation function. Higher order cumulants can be seen as a measure of gaussianility of a random process because cumulants higher than second order are zero in a gaussian process.

Bispectrum

Bispectrum is the Fourier transform of the third order cumulant. The bispectrum of a stationary Gaussian process with zero media are equal to zero. The bispectrum of a signal plus Gaussian noise is the same as that of the signal, whereas the power spectrum of a signal plus Gaussian noise is very different from the power spectrum of the signal alone.

Therefore, through bispectrum Gaussian noise can be separated from non-Gaussian noise and signal-to-noise ratios can be improved.

On the other hand, quadratic phase coupling can be detected and no minimum phase systems can be identified with the bispectrum.

Bicoherence

Closely related to the bispectrum is the third-order coherence measure, the bicoherence. Bicoherence is the bispectrum normalized.

Bicoherence is bounded between 0 and 1 values and it is used to detect quadratic phase coupling due to second order alinearities. A phase coupling between a linear combination of the frequency components ω_1 and ω_2 exists if the bicoherence has a value equal to one for a pair of frequencies (ω_1, ω_2).

Chaos Theory

The Chaos theory helps us to understand and interpret the observations from complex deterministic dynamical systems and it can be used to predict and control time series (Kantz & Schreiber, 1997). Until the appear-

ance of the chaos theory all irregular behaviour was interpreted as a stochastic behaviour and therefore unpredictable. Thanks to the chaos theory this is not necessarily true. For example, stochastic and chaotic systems have rich broadband power spectra and varying phase spectra. So, in order to distinguish between stochastic and chaotic systems the chaos theory is a powerful new tool.

A deterministic dynamical system describes the time evolution of a system in some phase space $\Gamma \in \Re^m$ (*m* dimensional vectorial space), where a state is specified by a vector $\bar{x} \in \Re^m$. This evolution can be expressed by ordinary differential equations (Kantz & Schreiber, 1997):

$$\frac{d}{dt}\bar{x}(t) = f(t, \bar{x}(t)), t \in \Re$$

or in discrete time t = nΔt by maps:

$$\bar{x}_{n+1} = F(\bar{x}_n), n \in Z$$

A sequence of points ($\bar{x}_n$ or $\bar{x}(t)$) that solve the equations of the system are called trajectories. The initial conditions are $\bar{x}_0$ or $\bar{x}(0)$, respectively. The region of the phase space in which all trajectories originated in a range of initial conditions converges after a transition time is called attractor. An example of a chaotic attractor from the Colpitts oscillator (Kennedy, 1994) is illustrated in Figure 1.

Most of the time we need to characterize nonlinear systems for which equations and models are unknown. However, some measurements of the system are known.

Figure 1. Attractor from Colpitts oscillator
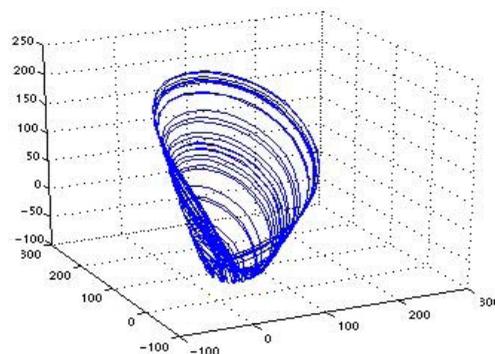
There exist some techniques to obtain the phase space and the attractor from the output signal (**embedding techniques**). Thus, certain quantities such as Lyapunov exponents, **correlation dimension** and Kolmogorov-Sinai entropy are obtained from the attractor. These quantities provide measurements of the nonlinearity degree of the system. These measurements are invariant under smooth transformations and thus independent of the embedding procedure.

Embedding Techniques

Takens' embedding theorem (Takens, 1981) states that an embedding exists if the dimension (m) of the reconstructed phase space is such that $m>2D+1$ (D is the attractor dimension). There exist two main methods to reconstruct the attractor from a time series: the **method of delays** (Kantz & Schreiber, 1997) and principal component analysis (Broomhead & King, 1986). The former method is the most popular: a delay reconstruction in m dimensions is formed by the vectors s_n given as (Kantz & Schreiber, 1997),

$$s_n = [s(n), s(n-T), ..., s(n-(m-1)T)]$$

where $s(n)$ is the scalar signal measured, m is the **embedding dimension** of the reconstructed phase space and T is the **time delay**.

Takens' theorem is strictly an existence theorem and does not suggest how to find the embedding dimension (m) and the time delay (T). The first zero of autocorrelation function or when it decays $\frac{1}{e}$ has been suggested as a first order **estimator** of T. The first minimum of mutual information function (Fraser & Swinney, 1986) is another estimator of T that takes into account nonlinear correlations.

The false neighbours method (Kennel, Brown & Abarbanel, 1992) and the false strands method are proposed methods to estimate the embedding dimension (m). The latter is an improvement of the false neighbours method.

Chaotic Measurements

In the following paragraphs some chaotic measurements will be described.

Lyapunov Exponents

Lyapunov exponents characterize the rate of separation of two points in phase space initially separated by a small distance. There exist as many Lyapunov exponents as m (dimension of the phase space). The maximal Lyapunov exponent (MLE) is the largest one and determines the predictability of a dynamical system. A positive MLE means divergence of nearby trajectories, i.e. chaos. For a mathematical description we refer the reader to (Kantz & Schreiber, 1997). Several algorithms to compute Lyapunov exponents from a time series have been implemented (Wolf et. al, 1985), (Rosenstein, Collins, De Luca, 1993), (Kantz, 1994), (Sprott, 2003).

MLE is useful to characterize different kinds of behaviour in a signal or system. A negative MLE is an indicator of a stable fixed point (a dissipative or non-conservative system), a positive MLE is an indicator of irregular (chaotic) behaviour, a zero MLE is an indicator of a conservative system (such as a harmonic oscillator) and an infinite MLE is an indicator of noise.

Kolmogorov-Sinai Entropy

Kolmogorov-Sinai (KS) entropy quantifies the loss of information as a system evolves and it is another measurement related to the unpredictability of a system. In a regular and predictable system, $H_{KS} = 0$, i.e. nearby points are closely grouped in some other small region of phase space and there is no change in information. In a random process $H_{KS} = \infty$ due to the fact that all phase space regions become possible after a short time. In chaotic systems $0 < H_{KS} < \infty$ indicates that nearby points in the phase space diverge exponentially. Therefore, according to KS entropy values different types of systems can be characterized: regular, chaotic and noise systems.

Correlation Dimension

Correlation dimension (Grassberger & Procaccia, 1983) quantifies the complexity of the reconstructed attractor. It is a geometric measurement of sensitive dependence on initial conditions because in chaotic motion the attractor usually shows a very complicated and fractal geometry. In a chaotic deterministic system the

correlation dimension yields to a finite value, whereas in a random process it does not converge to a value. A maximum likelihood **estimator** to obtain optimal values of correlation dimension is the Takens-Theiler estimator (Theiler, 1988).

Correlation dimension allows us to identify a random process from a chaotic motion. A non-integer (fractal) value of the correlation dimension is usually a symptom of chaos, whereas a integer value is a symptom of a regular behaviour. Furthermore, the correlation dimension is an estimation of the number of degrees of freedom of a system.

FUTURE TRENDS

In automatic recognition systems it is necessary to characterize data sequences and objects (voice, sounds, faces, hands, etc.) in order to achieve a well described features space. Having differential features will later lead to a successful classification process.

However, the task of finding differential features is not always easy. Nonlinear techniques are novel resources to characterize time series and overcome certain previous problems of linear techniques. Proof of this is the development of several automatic classification systems using nonlinear features such as (Alonso, de León, Alonso, Ferrer, 2001) (Alonso, Díaz-de-María, Travieso, Ferrer, 2005), (Ubeyli & Guler, 2004), (Gulera, Ubeylib & Guler, 2005).

CONCLUSION

In this article we have shown the state of the art in two recent nonlinear techniques: Higher order statistics and the chaos theory. The main point is the fact that many signals in real life cannot be adequately modelled by linear approximation alone. Recently, the development of packages to compute chaotic (TISEAN package, Hegger, Kantz & Schreiber, 1999) and HOS (HOSA toolbox for Matlab) measures from data sets has made the application of these techniques to data sets feasible.

Thanks to these techniques it is now possible to extract new characteristics previously ignored by linear analysis. Therefore the use of nonlinear techniques leads to more realistic characterization of signals and systems.

These new approaches to signal analysis and characterization provide new tools for the better characterization of signals and as a previous step in order to create new, more accurate and powerful automatic systems in patter recognition systems such as voice and facial recognition.

REFERENCES

Alonso, J.B., de León, J., Alonso, I. & Ferrer, M. A. (2001). Automatic detection of pathologies in the voice by hos based parameters. *EURASIP Journal on Applied Signal Processing*, 1, 275-284.

Alonso, J.B., Díaz-de-María, F., Travieso, C. M., Ferrer, M. A. (2005). Using Nonlinear Features for Voice Disorder Detection. *3rd International Conference on Nonlinear speech processing*, 94-106.

Broomhead, D. & King, G. (1986). Extracting qualitative dynamics from experimental data. *Physica D*, 20, 217–236.

Casdagli, M. (1989). Nonlinear prediction of chaotic time series. *Physica* D 35, 335.

Coroyer, C., Declercq, D., Duvaut, P. (1997). Texture classification using third order correlation tools. *IEEE Signal Processing Workshop on Higher-Order Statistics (SPW-HOS'97)*, p. 0171.

Fraser, A. M. & Swinney, H. L. (1986). Independent coordinates for strange attractors from mutual information, *Phys. Rev. A* 33, 1134-1140.

Grassberger, P. & Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D* 9, 189.

Gulera, N.F., Ubeylib, E.D. & Guler, I. (2005). Recurrent neural networks employing Lyapunov exponents for EEG signals classification. *Expert Systems with Applications,* 29, 506–514.

Harbourne, R. T., Stergiou, N. (2003). Nonlinear Analysis of the Development of Sitting Postural Control. *Wiley Periodicals*, *Inc.*

Hegger, R., Kantz, H. & Schreiber, T. (1999). *Practical implementation of nonlinear time series methods: The TISEAN package. Chaos* 9, 413.

Hommes, C.H. & Manzan, S. (2006). Testing for Nonlinear Structure and Chaos in Economic Time Series. *Tinbergen Institute Discussion Papers* No. 2006-030/1.

Kannathal, N., Lim Choo Min, Rajendra Acharya U. & Sadasivan, P.K. (2005). Entropies for detection of epilepsy in EEG. *Computer Methods and Programs in Biomedicine*, 80, 187–194.

Kantz, H. (1994). A robust method to estimate the maximal Lyapunov exponent of a time series. *Physics Letters* A, 185, 77-87.

Kantz, H. & Schreiber, T. (1997). Nonlinear Time Series Analysis. *Cambridge Nonlinear Science Series 7*.

Kennedy M. P. (1994), "Chaos in the Colpitts oscillator," IEEE Trans. Circ. Syst., vol. 41, pp. 771-774

Kennel, M. & Abarbanel, H. (2002). False neighbors and false strands: A reliable minimum embedding dimension algorithm. *Phys. Rev*. E 66.

Kennel, M., Brown, R., Abarbanel, H. (1992). Determining embedding dimension for phase space reconstruction using the method of false nearest neighbors. *Phys. Rev.* A 45, 3403 – 3411.

Logan, D., Mathew, J. (1996). Using the correlation dimension for vibration fault diagnosis of rolling element bearing – 2. Selection of experimental parameters. *Mechanical Systems and Signal Processing*, 10, 251-264.

Mendel, J.M. (1991). Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications. *IEEE, Proceedings*, 79, 278-305.

Nikias C.L. and Petropulu A.P. (1993), *Higher-Order Spectra analysis*, PTR Prentice Hall, New Jersey

Packard, N. H., Crutchfield, J. P., Farmer, J. D. & Shaw, R. S. (1980). Geometry from a Time Series. *Phys. Rev. Lett.* 45 (9), 712-716.

Rosenstein, M. T., Collins, J. J., De Luca, C. J. (1993). A practical method for calculating largest Lyapunov exponents from small data sets. *Physica* D 65, 117.

Ruelle, D. (1979). Sensitive dependence on initial condition and turbulent behaviour of dynamical systems. *Annals of the New York of Sciences* 316 (1), 408-416.

Samanta, B., Al.Balushi, K.R. & Al-Araimi, S.A. (2006). Artificial neural networks and genetic algorithm for bearing fault detection. Soft Computing, (10), 264-271.

Shen, M, Shen F. (1997).Time-varying third-order cumulant spectra and its application to the analysis and diagnosis of phonocardiogram. *IEEE Signal Processing Workshop on Higher-Order Statistics (SPW-HOS'97)*. p.0024.

Sprott, J. C. (2003). Chaos and Time-Series Analysis. Oxford, UK: Oxford University Press.

Takens, F. (1981). Detecting strange attractors in turbulence. *Lecture notes in mathematics. Dynamical systems and turbulence* (898), 366. Springer, Berlin.

Theiler, J. (1988).*Lacunarity in a best estimator of fractal dimension. Phys. Lett.* A 135, 195.

Tufillaro, N., Abbott, T. & Reilly, J. (1992). An expiremental Approach to Nonlinear Dynamics and Chaos. Reading, MA: Addison-Wesley.

Ubeyli, E.D. & Guler, I. (2004). Detection of electrocardiographic changes in partial epileptic patients using Lyapunov exponents with multilayer perceptron neural networks. *Engineering Applications of Artificial Intelligence, 6* (17), 567–576.

Van Zyl, J. (2001). Modelling Chaotic Systems with Neural Netwoks: Application to Seismic Event Predicting in Gold Mines. Thesis.

Wang, W.J. & Lin, R. M. (2003). The application of pseudo-phase portrait in machine condition monitoring. *Journal of Sound and Vibration*. 1 (259), 1-16.

Wolf, A., Swift, J.B., Swinney, H.L. & Vastano, J.A. (1985). Determining Lyapunov exponents from a time series. *Physica* D, 16, 285-317.

KEY TERMS

Attractor: A region in the phase space to which all trajectories converge after a transition time. It is the long term behaviour of a dynamical system.

Bicoherence: It is a normalised version of the bispectrum. The bicoherence takes values bounded

between 0 and 1, which make it a convenient measure for quantifying the phase coupling in a signal.

Chaos: Long-term unpredictable behaviour caused by sensitive dependence on initial conditions.

Cumulants: The kth order cumulant is a function of the moments of orders up to and including k.

HOS: Higher order statistics is a field of statistical signal processing that uses more information than autocorrelation functions and spectrum. It uses moments, cumulants and polyspectra. They can be used to get better estimates of parameters in noisy situations, or to detect nonlinearities in the signals.

Kolmogorov-Sinai Entropy: Measurement of information loss per unit of time in phase space.

Lyapunov Exponents: Quantity that characterizes the rate of separation of infinitesimally close trajectories in a dynamical system. The maximal Lyapunov exponent (MLE) determines the predictability of a dynamical system. A positive MLE means a chaotic system.

Polyespectra: The Fourier transform of cumulants. The second order polyspectra is the power spectrum. Most HOS work on polyspectra focusses attention on the bispectrum and the trispectrum.

Reconstructed Phase Space: Phase space obtained from a time series through embedding techniques such as principal component analysis or the method of delays.

Ontologies and Processing Patterns for Microarrays

Mónica Miguélez Rico
University of A Coruña, Spain

José Antonio Seoane Fernández
University of A Coruña, Spain

Julián Dorado de la Calle
University of A Coruña, Spain

INTRODUCTION

The researchers currently have a new tool for dealing with the solution of biomedical problems: the Microarrays. These devices support the study and the acquisition of information related to many genes at the same time by means of a unique experiment, providing multiple potential applications such as mutation detection of microorganism identification.

Some of the problems that exist when working with this type of technologies are the high number of data and the complex technical nomenclature to be dealt with. These facts imply the need of using several standards and ontologies when performing this type of experiments.

BACKGROUND

The microarrays have been a key element in the bio-technological revolution of the last years; however new problems regarding both, data handling and statistics analysis, have arisen due to the vast volume of information and to the structure of the data used.

The main concern lies in the vast amount of data to be stored, processed and analysed. Besides, as the microarrays are a new technique, most of the methods, protocols and standards are still being defined.

The fact of dealing with such amount of unstructured information leads to believe that is quite difficult for the descriptors of the stored concepts or their units to be the same at the different data bases where it is accessed. In order to support the vocabulary unification task, the ontologies (Chandrasekaran, 1999) enable a hierarchical definition of concepts for framing the schemas of the accessed data bases. There are fully established ontologies also quite used as the UMLS medical vocabulary (UMLS, 2006), that has information about symptoms and illnesses, or the GO (Gene Ontology) genomic ontology (Gene Ontology, 2006), regarding information about the function and the expression location of the different human genes.

Once the use of ontologies has been established, they are also quite useful for searching hidden relationships among data. Consultations with SQL-type (Structured Query Language) (Beaulieu, 2005) query languages may be performed in an ontology and translated to query languages owning to each underlying data base. In this way, by the use of the ontology, it could be known that the presence of fever is a symptom and which are the illnesses that present fever as a symptom.

Currently, there are special data formats in medicine science as the DICOM standard (Oosterwijk, 2001) for storage and transfer of the increasing amount of medical images that support new imaging modalities. Nevertheless, the typical biomedical images, as the microarrays or the DNA gels, are not currently considered at DICOM, although their future integration is foreseeable in incoming revisions, as the clinical test based on these techniques might be increasingly used in routine medical practice. At the moment, however, the management of this type of images is quite sensitive.

MAIN FOCUS OF THE CHAPTER

This paper presents a description of the most important standards and ontologies for working with microarrays

experiments; it also tackles the integration options of some of these ontologies and standards into an information system for managing microarrays.

The first standardisation initiatives appeared in 1998. They were more or less isolated initiatives where three standardisation areas could be distinguished: hardware, fixed material and procedures for analysis and storage of studies information. Several organisations as the MGED Normalization Working Group (MGED Data, 2006) were created for the standardisation of the information. The MGED (Microarray Gene Expression Data) Society is an international organisation devoted to the standardisation and to the exchange of information related to microarrays experiments. Other organisations to be mentioned are the OMG (Object Management Group) (OMG, 2006) or the UCL/HGNC (Human Gene Nomenclature) (HGNC, 2006).

As far as terminologies, vocabularies, nomenclatures and ontologies is concerned, it should be highlighted the MGED Ontology (MGED OWG, 2006), which describes the experiments and the gene expression data, or the GO (Gen Ontology Consortium) (Gene Ontology, 2006), which provides controlled vocabularies for describing the molecular function, the biological process and the cellular components of the gene products. Also the UCL/HGNC (Human Gene Nomenclature) (HGNC, 2006), the TaO (TAMBIS Ontology) (TaO, 2006), the

RiboWeb (RiboWeb, 2001) or the EcoCyc (EcoCyc, 2005) should be mentioned.

Regarding the data exchange standards in the microarrays field, the MicroArray and Gene Expression Markup Language (MAGE-ML) (MAGE-ML, 2006) is language designed for describing and communicating information among microarrays experiments.

Other data exchange standards are the Bioinformatics Sequence Markup Language (BSML) (BSML, 2006), the Gene Expression Markup Language (GeneXML) (NCGR, 2006) or the Genome Annotation Markup Elements (GAME) (Bioxml, 2006).

The MGED Group is the standardisation organisation that presents the wider scope regarding the microarrays field and presented in November 2000 the standard **MIAME (Minimun Information About a Microarray Experiment)** (MIAME, 2006). This acronym describe the minimal information regarding microarrays that, either should be stored into a data base (from now, DD.BB) used as a public repository, or that should be stored for enabling the non ambiguous interpretation of the experiments results and for repeating such experiments.

After defining the information that is going to be stored (MIAME), there should be a model of objects (UML) for describing, not only how the data of these experiments should be expressed, but also the mecha-

Figure 1. MGED ontology

nisms for their exchange, bearing always in mind the MIAME guides. This is precisely what the **MAGE-OM (MicroArray and Gene Experiment Object Model)** (MAGE-OM, 2006) standard defines.

This model of objects has been developed for being independently used from the implementation chosen and, in this way, it can be used as a map for data structures in platforms such as Java, Perl or C++. The model has been currently translated to a set of relational tables divided in packages, according to the natural separation of the gene expression data into cases and objects.

In this point, and by the use of standards already described, the microarrays experiments data to be stored and their model of objects are both defined. A language for the data exchange is therefore needed, as the **MAGE-ML (MicroArray Gene Expression Markup Language** (MAGE-ML, 2006).

It is a XML (XML, 2006) formal language directly derived from the MAGE-OM object model. This language has been designed for describing and communicating the information of such type of elements and it can be used for describing microarrays-related items such the designs, information about the fabrication or the structure of experiments.

A tool named as **MAGE-stk (MAGE Software Toolkit)** has been developed in order to simplify the use of the MAGE-OM standard. This tool is based on an Open Source package collection that implements the MAGE model of objects (MAGE-OM) in several programming languages. It makes the reading of the MAGE-ML easier; this tool also simplifies the MAGE-ML writing from MAGE-OM and it provides methods for the fully maintenance, as well as actualisation, of MAGE-OM.

Once the standards needed for working with microarrays technology have been defined, the following step is the description and the use of several ontologies that might enable, as it was mentioned before, the unification of the different vocabularies used.

The **MGED Ontology (MO)** is one of the most important ontologies when working with microarrays and, particularly, when using certain previously mentioned standards. The main goal of this ontology is to provide standard terms for the notation of experiments with microarrays; such terms not only will serve for structuring questions related to the elements of the experiments, but also they might be used for unambiguously describing how the experiments have been done.

As the ontology-encoded terms will be eventually placed in MAGE-ML documents, the efforts of both, MAGE and the working group, should be coordinated at the points where they superimpose, for the ontology classes and the MAGE classes to have the same names and relationships.

The ontology has been conceived for continuously growing and therefore fulfilling the requirements of descriptive terms related to emerging applications of microarrays. Besides, the use of ontologies for software programming should be fixed, in order to avoid constant revisions of the programming for searching changes in vocabularies and relationships. The fulfilment of such objectives is achieved by establishing the central MGED ontology, a nucleus at the MGED ontology that will remain constant. The extended MGED ontology is a second ontology layer that contains all the additional terms that might be considered (see Figure 1).

The central MGED ontology has been developed for working with the MAGE 1.0 schema, and it is restricted to MAGE-OM v1.1. The extended MGED ontology increases the ontology nucleus with terms that are out of reach of MAGE v1.1.

The **Gen Ontology (GO)** is other ontology that should be considered when working with microarrays. The Gen Ontology Project implies a collaborative effort in order to fulfil the needs of consistent descriptors for genetic products in different DD.BB. The project started in 1998 as collaboration among three DD.BB. related to models of organisms: FlyBase, Saccharomyces Genome Database (SGD) and the Mouse Genome Database (MGD). Since then, the GO consortium has grown and includes many more DD.BB., as some of the world biggest repositories for plants, animals and microbial genomes.

The GO project has developed three controlled and structured ontologies/vocabularies: biological processes, molecular functions and cellular components. In this way, a given gene can be located in one or more *cellular components*, the *biological processes* where it is active can be checked and the *molecular functions* represented by that gene at those processes can be visualised. For instance, the 'cytochrome c' gene can be described by the *molecular function* term 'oxidoreducta activity', by the *biological process* terms 'oxidate phosphorylation' and 'death cell induction' and by the *cellular component* terms 'mitochondrial matrix' and 'mitochondrial membrane'.

ESSENTIAL CHARACTERISTICS OF AN INFORMATION SYSTEM FOR MICROARRAYS MANAGEMENT

This type of system needs an architecture of data integration for easily store the vast amount of information generated by the experiments with microarrays; In order to achieve this, the architecture should provide the users with assistants and contextual support for handling the information. On the other hand, a Web architecture, by means of an Internet connection, will enable the access and the management of the information from any place at any time.

For the ontology information to be always actualised and available for the users, the architecture should provide an integrated access to several ontology servers. In order to achieve this, it should be advisable to use Web Services in cases as the access to the Gen Ontology (GO) or to the Biological Imaging Methods (FBbi); alternatively, Internet access should be used in MGED Ontology access. Besides, for the users to introduce data and consult the stored information more easily, the system should have an interface that might show a list of ontology terms and values; in this way, this list would enable ontology consultations that might include all the meanings of a given concept.

As the proposed system has to support the information exchange among the different researchers, this type of architecture should use the existing standards related to data storage (MIAME) and to information exchange (MAGE-OM y MAGE-ML). In the first case, the system should have to implement a DD.BB. whose fields fulfil the MIAME standard; in the second case, the system will use the MAGE-OM object model for enabling the generation of the MAGE-ML information exchange file by the users whenever they might require it.

Lastly, it should be also advisable that the users could continue using the existing applications, to which they are used to, and that have been developed by experts on the subject usually using the R language. Due to that reason, the system should have such applications available for the users. In order to achieve this, it is proposed an approach based in the use of Web services by the architecture.

This architecture is being currently developed by the RNASA/IMEDIR lab group from the University of A Coruña.

CONCLUSION

Nowadays there are several tools that enable the analysis of microarrays imaging; however, as they are software specifically designed for each array type, they do not allow wide options and they, not only require to be installed in the user machine, but also its installation is restricted to a few operative systems.

Regarding data processing, there are several projects that include packages for performing microarrays imaging processings as normalisation or clustering; however, some of these packages need to download the different processing tools that they contain in order to use them.

Lastly, there are several types of public DD.BB. for storing the information of this type of experiments by the use of Web formularies. As there are also some stand-alone tools that store the data into a DD.BB. created in the machine of the user, this machine should have a DD.BB. manager installed.

Nevertheless, no systems have been found to perform the different steps without needing to install software or to quit the system.

The new systems of this area should allow the data storage into a MIAME standard DD.BB. with the option of performing the image analysis of the different microarrays experiments and keeping the analysis results into de system DD.BB. The systems should also provide several processing types using R language in order to perform data analysis and subsequent experiment conclusions. The data model of the system should use MAGE-OM standard and then offer the resulting experiment MAGE-ML file to the user.

ACKNOWLEDGMENT

REFERENCES

Beaulieu, A. (2005). Learning SQL. O'Reilly.

Bioxml. (2006). Consulted in July 2006 from, http://www.bioxml.org/Projects/game/

BSML. Bioinformatics Sequence Markup Language. (2006). Consulted in June 2006, from http://www.bsml.org/

Chandrasekaran, B., Josephson, J. R. & Richard, V. (1999). What Are Ontologies, and Why Do We Need Them?. IEEE Intelligent Systems. 14 (1), 20 – 26.

EcoCyc. (2005). Consulted in February 2006, from http://ecocyc.org/

Gene Ontology. Consulted in September 2006, from http://www.geneontology.org/

HGNC. HUGO Gene Nomenclature Committee. (2006). Consulted in March 2006, from http://www.gene.ucl.ac.uk/nomenclature/

MAGE-ML. (2006). Consulted in May 2006, from http://www.mged.org/Workgroups/MAGE/mage-ml.html

MAGE-OM. (2006). Consulted in May 2006, from http://www.mged.org/Workgroups/MAGE/mage-om.html

MGED Data Transformation an Normalization Working Group. (2006). Consulted in June 2006, from http://genome-www5.stanford.edu/mged/normalization.html

MGED OWG. The MGED Ontology is an experimental Ontology. Consulted in August 2006, from http://**mged**.sourceforge.net/ontologies/Ontology-WorkshopMGED6.ppt

MIAME (2006). Consulted in July 2006, from http://www.mged.org/Workgroups/MIAME/miame.html

NCGR. (2006). Consulted in June 2006, from http://www.ncgr.org/genex/

OMG. Object Management Group. (2006). Consulted in March 2006, from http://www.omg.org/

Oosterwijk, H. (2001). DICOM Básico. 2ª Edición. OTech.

RiboWeb. (2001). Consulted in February 2006, from http://riboweb.stanford.edu/

TaO: TAMBIS Ontology. (2006). Consulted in February 2006, from http://imgproj.cs.man.ac.uk/tambis/

Unified Medical Language System. in February 2006, from http://www.nlm.nih.gov/research/umls/.

Extensible Markup Language. (2006). in August 2006, from http://www.w3.org/XML/

KEY TERMS

MAGE-ML: *Microarray Gene Expression Markup Language*. Formal language designed for describing and communicating the experiment-based microarrays information.

MAGE-OM: *MicroArray and Gene Experiment Object Model*. Standard that defines the model of objects for the gene expression-based experiments.

MAGE-stk: MAGE Software Toolkit. Open Source Package collection that implements the MAGE (MAGE-OM) model of objects in several programming languages.

MIAME: *Minimum Information About a Microarray Experiment*. Standard that indicates the minimal information needed for microarrays experiments.

MicroArrays: A technology using a high-density array of nucleic acids, protein, or tissue for simultaneously examining complex biological interactions which are identified by specific location on a slide array. A scanning microscope detects the bound, labelled sample and measures the visualized probe to ascertain the activity of the genes of interest in genotyping, cellular studies, and expression analysis.

Ontology: In computer science this term refers to the attempt of formulate an exhaustive and rigorous conceptual schema into a given domain, with the aim of making communication and information sharing among systems easier.

R: Language and programming environment for graphic and statistical analysis.

O

Ontologies for Education and Learning Design

Manuel Lama
University of Santiago de Compostela, Spain

Eduardo Sánchez
University of Santiago de Compostela, Spain

INTRODUCTION

In the last years, the growing of the Internet have opened the door to new ways of learning and education methodologies. Furthermore, the appearance of different tools and applications has increased the need for interoperable as well as reusable learning contents, teaching resources and educational tools (Wiley, 2000). Driven by this new environment, several metadata specifications describing learning resources, such as IEEE LOM (LTCS, 2002) or Dublin Core (DCMI, 2004), and learning design processes (Rawlings et al., 2002) have appeared. In this context, the term learning design is used to describe the method that enables learners to achieve learning objectives after a set of activities are carried out using the resources of an environment. From the proposed specifications, the IMS (IMS, 2003) has emerged as the de facto standard that facilitates the representation of any learning design that can be based on a wide range of pedagogical techniques.

The metadata specifications are useful solutions to describe educational resources in order to favour the interoperability and reuse between learning software platforms. However, the majority of the metadata standards are just focused on determining the vocabulary to represent the different aspects of the learning process, while the meaning of the metadata elements is usually described in natural language. Although this description is easy to understand for the learning participants, it is not appropriate for software programs designed to process the metadata. To solve this issue, ontologies (Gómez-Pérez, Fernández-López, and Corcho, 2004) could be used to describe formally and explicitly the structure and meaning of the metadata elements; that is, an ontology would semantically describe the metadata concepts. Furthermore, both metadata and ontologies emphasize that its description must be shared (or standardized) for a given community.

In this paper, we present a short review of the main ontologies developed in last years in the Education field, focusing on the use that authors have given to the ontologies. As we will show, ontologies solve issues related with the inconsistencies of using natural language descriptions and with the consensus for managing the semantics of a given specification.

ONTOLOGIES IN EDUCATION

In the educational domain a number of ontologies have been developed for authors. Thus ontologies have been developed to describe the learning contents of technical documents and formalize the semantics of learning objects; model the elements required for the design, analysis, and evaluation of the interaction between learners in computer supported cooperative learning; and describe the learning design associated to a unit of learning in which the learning flow is explicitly declared.

Ontologies in Learning Contents and Metadata

The main purpose of these ontologies is to describe the contents or features of documents in order to favor its indexing and retrieval from applications. Thus Kabel, Wielinga, and Hoog (1999) develop three ontologies that annotate technical documents from a given domain: these documents are converted in a large collection of information elements described by a number of attributes to which values are assigned from the ontologies. These attributes are referred to the subject matter in the application domain, structural and representational properties (paragraphs, sections, etc.) and the potential instructional roles of the information elements. Following this approach the ontologies represent the

semantics of the documents, enabling its indexing and retrieving from databases.

Other interesting ontology in this field is proposed by Brase, Painter and Nejdl (2004). Using an ontology language as TRIPLE, this ontology describes the semantics of the LOM specification, adding formal axioms and rules to the metadata representation of the standard. With this formal description the semantics of the LOM specification is not changed, but it helps to define the constraints on LOM fields, making clear the meaning and use of these LOM fields, resulting in easier exchange of LOM metadata between different applications and contexts.

Ontologies in Collaborative Learning Environments

These ontologies are used to model the interaction between the learning actors (typically teachers and students) in collaborative environments. Thus Inaba et al. (2001) present an ontology a collaborative learning ontology that facilitates the design, analysis, and evaluation of a collaborative learning sesion. This ontology describes the concepts of several well-established learning theories, defining the semantics of what learning goal concept is and connecting this concept with the theories which are formulated in a taxonomy. In this work, authors have used the ontology to facilitate users the design and execution of the instructional process in a collaborative environment (Barros, Verdejo, Read, & Mizoguchi, 2002).

Ontologies in Learning Design

These ontologies focus on the semantic description of the learning design modelling which defines the learning flow of the activities to be carried out by teachers and students. The ontologies developed in this field are based on the IMS Learning Design (IMS LD) specification which has risen as a de facto standard for defining learning designs. This specification has: (1) a well-founded conceptual model that declares the vocabulary and the functional relations between the concepts of the learning design; (2) an information model that describes in an informal (natural language) way the semantics of every concept and relation introduced in the conceptual model; and (3) a behavioural model that specifies the constraints imposed to the software system when a given learning deisgn is executed in

runtime. In other words, the behavioural model defines the semantics of the IMS LD specification during the execution phase. Figure 1 depicts the main concepts of the IMS LD specification.

Knight, Gasevic and Richards (2006) present a general framework whose prupose is to save the gap between learning designs and the learning objects used in them. For achieved this, the framework considers the development of three ontologies that describe the learning design, the learning objects and the context in which these objects are used. LOCO is the ontology, defined in the language OWL (Dean & Schreiber, 2004), that deals with the description of learning designs. It represents the semantics specified in IMS LD and, particularly, in its conceptual model, which means that LOCO integrates the concepts and relations defined in the conceptual and information models of the IMS

Figure 1. Main concepts of the IMS Learning Design specification (Amorim et al., 2006)
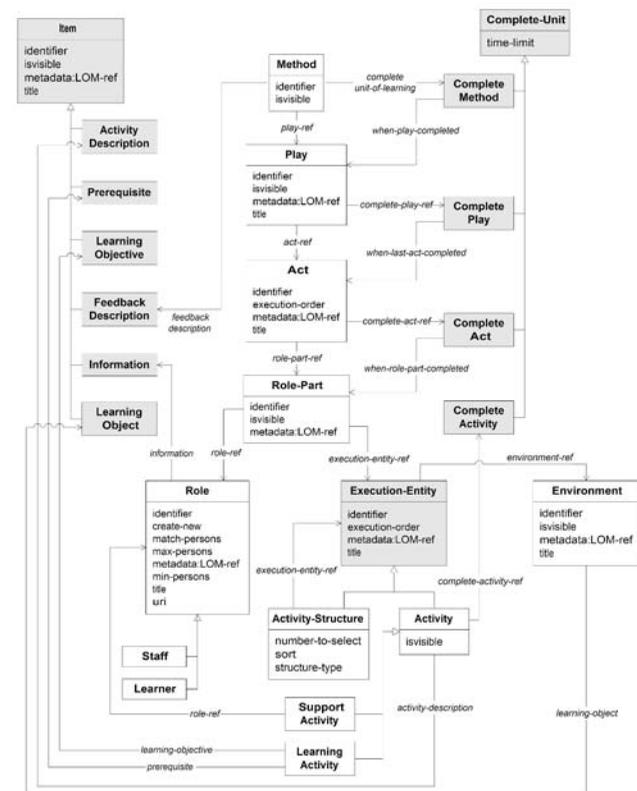
Table 1. Examples of axioms that constrain the semantics of the IMS LD concepts

Design Axiom 1	IMS LD Specification	*Page 38 (item 0.2.2)*: "The time limit specifies that it is completed when a certain amount of time has passed, relative to the start of the run of the current unit of learning. The time is always counted relative to the time when the run of the unit-of-learning has been started. Authors have to take care that the time limits set on role-parts, acts and plays are logical."
	Explanation	The value of the attribute `time limit` of a `Method` must be greater than the value of the `time limit` of any `Play`. That is, the `Play(s)` cannot finish after the `Method`.
	Formal Description	$\forall$ m, p, cm, cp $\mid$ m $\in$ Method $\wedge$ p $\in$ Play $\wedge$ cm $\in$ Complete-Method $\wedge$ cp $\in$ Complete-Play $\wedge$ play-ref(p, m) $\wedge$ complete-unit-of-learning-ref(cm, m) $\wedge$ complete-play-ref(cp, p) time-limit(cm) $\geq$ time-limit(cp)
Design Axiom 2	IMS LD Specification	*Page 90*: "The same role can be associated with different activities or environments in different role-parts, and the same activity or environment can be associated with different roles in different role-parts. However, the same role may only be referenced once in the same act."
	Explanation	For the same `Act`, the `Roles` involved in the execution of the `Act` are disjoint.
	Formal Description	$\forall$ a, r, rp $\mid$ a $\in$ Act $\wedge$ r $\in$ Role $\wedge$ rp $\in$ Role-Part $\wedge$ role-part-ref(rp, a) $\wedge$ role-ref(r, rp) $\neg$ $\exists$ rp1 $\mid$ rp1 $\in$ Role-Part $\wedge$ rp1 $\neq$ rp $\wedge$ role-part-ref(rp1, a) $\wedge$ role-ref(r, rp1)
Runtime Axiom 1	IMS LD Specification	*Page 25 (item 0.2.1)*: "The create-new attribute indicates whether multiple occurrences of this role may be created during runtime. When the attribute has the value "not-allowed" then there is always one and only one instance of the role."
	Explanation	If the value of the attribute `create-new` is "not-allowed", it can have an only instance of the `Role` at which it is applied.
	Formal Description	$\forall$ r $\mid$ r $\in$ Role $\wedge$ create-new(r) = "not-allowed" $\neg$ $\exists$ r1 $\mid$ r1 $\in$ r

LD standard, but the semantics expressed in natural language is not included in the ontology.

To deal with this issue, Amorim, Lama, Sánchez, Riera and Vila (2006) propose an ontology also based on the IMS LD that incorporates all its semantics, adding a number of axioms to the conceptual model: they are extracted from the information model where are expressed as natural language restrictions to the values of the concept attributes (table 1). Therefore this ontology does not modify the IMS LD spefication, but it incorporates all the semantics in order to enable software programs to manage directly from the representation in the ontology. With this formal specification this ontology, which is developed in F-Logic (Kiefer, Lausen, Wu, 1996) and OWL, has been used to validate the consistency of unit of learnings defined in authoring tools and as a language for knowledge interchanging between agents in collaborative environment (Riera et al., 2005).

CONCLUSION

Ontologies in Education are usually developed following a metadata standard whose intend is capture the semantics of a given theory or specification. Most of metadata standards have been modelled following the XML-Schema language (Thompson, Beech, Maloney, & Mendelsohn, 2004) which is not expressive enough to describe the semantics (or meaning) associated to the elements defined in the metadata. Thus, the main limitations of the XML-Schema language are (Gil & Ratnakar, 2002) that hierarchical relations between two or more concepts cannot be explicitly defined, and general and formal constraints (or axioms) between concepts, attributes, and relations cannot be specified.

To solve these limitations of the XML-Schema language the modelling of metadata standards needs to be enriched in order to describe explicitly and formally the semantics of its elements. Thus misinterpretations or errors are avoided when the instances of the concepts are created. This is the main purpose of the ontologies

developed in the Education field: to favour the interoperability between software programs by representing all the semantics of the metadata, not only the concepts and relations expressed in XML-based formats.

ACKNOWLEDGMENT

REFERENCES

Amorim, R., Lama, M., Sánchez, E., Riera, A., & Vila, X.A. (2006). A learning design ontology based on the IMS specification. *Journal of Educational Technology and Society*, 9(1), 38-57.

Barros, B., Verdejo, F., Read, T., & Mizoguchi, R. (2002). Applications of a Collaborative Learning Ontology. In C.A. Coello, A. de Albornoz, L.E. Sucar, & O.C. Battistutti (Ed.), *Proceedings of the Second Mexican International Conference on Artificial Intelligence* (pp. 301-310), Yucatan, Mexico.

Brase, J., & Nejdl, W. (2004). Ontologies and Metadata for eLearning. In S. Staab & R. Studer (Ed.), *Handbook on Ontologies* (pp. 555-574). Berlin: Springer-Verlag.

Dean, M., & Schreiber, G. (editors) (2004). *OWL – Web Ontology Language Reference. W3C Recommendation*. http://www.w3.org/TR/owl-ref.

Dublin Core Metadata Initiative (2004). *Dublin Core Metadata Element Set, Version 1.1. Reference Description*. http://dublincore.org/documents/dces.

Gil, Y., & Ratnakar, V. (2002). A Comparison of (Semantic) Markup Languages. In S.M. Haller, & G. Simmons (Eds.), *Proceedings of the Fifteenth International FLAIRS Conference* (pp. 413-418), Pensacola Beach, Florida.

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). *Ontological Engineering*. Berlin: Springer Verlag.

IEEE Learning Technology Standards Committee (2002). *Draft Standard for Learning Object Metadata (LOM)*. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

IMS Global Learning Consortium (2003). *IMS Learning Design Information Model. Version 1.0 Final Specification*. http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html

Inaba, A., Tamura, T., Ohkubo, R., Ikeda, M., Mizoguchi, R., & Toyoda, J. (2001). Design and Analysis of Learners Interaction based on Collaborative Learning Ontology. *Proceedings of the Second European Conference on Computer-Supported Collaborative Learning (Euro-CSCL'2001)* (pp. 308-315).

Kabel, S., Wielinga, B., & de Hoog, R. (1999). Ontologies for indexing Technical Manuals for Instruction. *Proceedings of the AIED-Workshop on Ontologies for Intelligent Educational Systems* (pp. 44-53), LeMans, France.

Kifer, M., Lausen, G., and Wu, J. (1995). Logical foundations of object oriented and frame based languages. *Journal of ACM*, 42, 741-843.

Riera, A., Sánchez, E., Lama, M., Amorim, R., Vila, X., & Barro, S. (2004). Study of Communication in a Multi-Agent System for Collaborative Learning Scenarios. *Proceedings of the Twelfth Euromicro Conference on Parallel, Distributed and Network based Processing* (pp. 233-240), A Coruña, Spain.

Rawlings, A., Rosmalen, P., Koper, R., Rodríguez-Artacho, M., & P. Lefrere (2002). Survey of Educational Modelling Languages (EMLs). *CEN/ISSS WS/LT Learning Techonologies Workshop*.

Sintek, M., & Decker, S. (2002). TRIPLE---A Query, Inference, and Transformation Language for the Semantic Web. In I. Horrocks, & J.A. Hendler. *Proceedings of the International Semantic Web Conference*, Sardinia, Italy.

Thompson, H., Beech, D., Maloney, M., & Mendelsohn, N. (2004). *XML-Schema Part 1: Structures Second Edition*. http://www.w3.org/TR/xmlschema-1

Wiley, D. (2000). *Learning Object Design and Sequencing Theory*. Department of Instructional Psychology and Technology. Brigham Young University. Doctoral Thesis.

KEY TERMS

Collaborative Learning Environment: Software system oriented to support collaborative learning experience in which two or more agents engage the goal of constructing knowledge based on group discussion and decision-making processes.

Interoperability: Capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.

Learning Design: Description of a method enabling learners to attain certain learning objectives by performing certain learning activities in a certain order in the context of a certain learning environment. A learning design is based on the pedagogical principles of the designer and on specific domain and contexts variables (e.g., designs for math be ematics teaching can differ from designs for language teaching).

Learning Objects: Any reproducible and addressable digital or non-digital resource used to perform learning activities or support activities. Examples are: web pages, text books, text processors, instruments, etc.

Metadata: Information about data, which can be used to comprehend, use, and manage data.

Ontology: Formal and explicit specification of a shared conceptualization, where *conceptualization* refers to an abstract model of a concept in the world; *formal* means that the ontology should be machine readable; *explicit* means that the type of concepts and the constraints on their use are explicitly defined; and *shared* reflects the notion that an ontology captures consensual knowledge accepted by a group.

Ontology Language: Formal language based on a logic paradimg that can represent concepts and the constraints between them. Reasoning capabilities of the language depend on the paradigm in which the language is based on.

Ontology Alignment Overview

José Manuel Vázquez Naya
University of A Coruña, Spain

Marcos Martínez Romero
University of A Coruña, Spain

Javier Pereira Loureiro
University of A Coruña, Spain

Alejandro Pazos Sierra
University of A Coruña, Spain

INTRODUCTION

At present, ontologies are considered to be an appropriate solution to the problem of heterogeneity in data, since ontological methods make it possible to reach a common understanding of concepts in a particular domain. However, utilizing a single ontology is neither always possible nor recommendable, given that different tasks or different points of view usually require different conceptualizations. This can lead to the usage of different ontologies, although in some cases the different ontologies collectively might contain information that could be overlapping and possibly even contradictory. This, in turn, represents another type of heterogeneity that can result in inefficient processing or misinterpretation of data, information, and knowledge.

To address this problem while at the same time insure an appropriate level of interoperability between heterogeneous systems, it is necessary to find correspondences or mappings that exist between the elements of the (different) ontologies being used. This process is known as ontology alignment.

This article offers an updated overview of ontology alignment, including a detailed explanation of what alignment consists of, and how it can be achieved. First, ontologies are defined using a fusion of different interpretations. This is followed by a definition of the concept of ontology alignment and, using a simple example, some of the most commonly used alignment techniques are illustrated. Subsequently, a case is made for the importance of automating the process of ontology alignment, summarizing some of the main alignment systems currently in use. Finally, in the context of future directions, a discussion is presented of the advantages associated with integrating ontology alignment into systems that require exchanging information in an automatic fashion.

BACKGROUND

Towards the end of the 20th and beginning of the 21st centuries, the term "ontology" (or ontologies) gained usage in computer science to refer to a research area in the subfield of artificial intelligence primarily concerned with the semantics of concepts and with expressive (or interpretive) processes in computer-based communications. In this context, there are many definitions of ontology, and these definitions have evolved over the years. Gruber offered one of the first definitions of ontology in 1993, as follows (Gruber, 1993):

"An ontology is an explicit specification of a conceptualization".

Gruber's definition became the most frequently referenced one in the literature, and became the base or working definition for those working in this area.

At present, ontologies are viewed as a practical way to conceptualize information that is expressed in electronic format, and are being used in many applications including the Semantic Web, e-Commerce, data warehouses, or information integration and retrieval. The basic idea behind these applications is to use ontologies to reach a common level of understanding or comprehension within a particular domain (e.g., a particular industry, medicine, housing, car repair, finances, etc.).

However, certain systems that encompass a large number of components associated with different domains would generally require the use of different ontologies. In such cases, using ontologies would not reduce heterogeneity but rather would recast the heterogeneity problem into a different (and higher) framework wherein the problem becomes one of ontology alignment, thereby allowing a more efficient exchange of information and knowledge derived from different (heterogeneous) data bases, knowledge bases, and the knowledge contained in the ontologies themselves. In this manner, ontology alignment enhances system interoperability.

ONTOLOGY ALIGNMENT

Euzenat et al. defined the problem of ontology alignment in the following manner (Euzenat et al., 2004):

"Given two ontologies which describe each a set of discrete entities (which can be classes, properties, rules, predicates, etc.), find the relationships (e.g. equivalence or subsumption) holding between these entities."

The key issue in ontology alignment is finding which entity in one ontology corresponds (in terms of meaning) to another entity in one (or many) ontology (or ontologies). Essentially, one might say that ontology alignment can be reduced to defining a similarity measure between entities in different ontologies and selecting a set of correspondences between entities of different ontologies with the highest similarity measures.

There are different methods to calculate the similarity measures between entities, and collectively these methods are known as **ontology alignment techniques**. Many of these techniques are derived from other fields (for instance, discrete mathematics, automatic learning, data base design, pattern recognition, among others). Consequently, some of these techniques attempt to compare text strings that describe the entities in the ontologies (terminology-based ontology alignment), while others calculate the similarity measures between entities taking into account the structure of their corresponding ontologies (structural ontology alignment). A complete classification of alignment techniques has been developed by Martínez (Martínez, 2007).

Using a simple example, the following discussion illustrates some of the basic ontology alignment techniques that are currently used. In this example, two simple ontologies are examined, as shown in Figure 1.

The ontologies shown in Figure 1 describe various entities in the real world: sets of elements that share certain characteristics or *classes* (e.g., *Wing*, *Car*, *Bus*, etc.), *instances* of classes (*individuals*) and their *relations* (e.g. a specific *Ferrari F50* belongs to a

Figure 1. An example illustrating the alignment between two ontologies
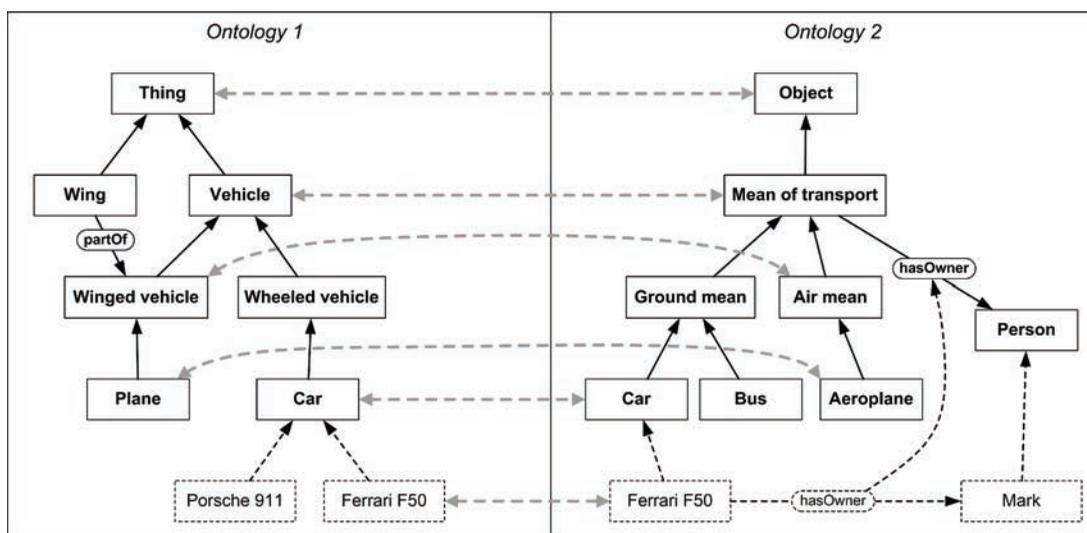
Table 1. Some examples of ontology alignment techniques

Correspondence	Technique Used	Description
Thing – Object *Vehicle – Mean of transport*	Language-based terminological technique	A support tool such as a dictionary is used (e.g. WordNet, 2007) to uncover that both terms are synonymous.
Car – Car *Ferrari F50 – Ferrari F50*	Terminological technique based on text strings	Text string that describe the entities completely coincide, since it can be shown that both entities have the same or similar semantics.
Plane – Aeroplane	Terminological technique based on text strings (suffix)	The first term is a suffix of the second, which would indicate that a relationship exists between them.
Winged vehicle – Air mean	Structural technique	In the first ontology, *Winged vehicle* is a child class of *Vehicle* and parent class of *Plane*. In the second, *Air mean* is child class of *Mean of transport* and parent class of *Aeroplane*. Since *Vehicle* was shown to be equivalent to *Mean of transport,* and *Plane* refers to the same concept as *Aeroplane*, both classes would show ascendants and descendants of the same or similar semantics, indicating a semantic relationships between them.

specific person, *Mark*), as well as three different types of *relationships* between individuals (*isA, partOf* and *hasOwner*).

Each one of the ontologies presented in this example has its own set of entities organized according to a specific taxonomy. The two representations arise due to the fact that they correspond to two different perspectives or points of view, each associated with a different domain. However, some pairs of entities can be identified in these ontologies that share the same or similar semantics. Thus, it's probable that the *Plane* class in the first ontology and *Aeroplane* in the second ontology refer to the same concept in general (in the real world), given that the terms that describe them are synonymous terms. Table 1 shows some of the pairs of entities of these ontologies among which semantic similarities could exist, as would be revealed once alignment techniques are applied. The technique that is being applied in each case is shown, along with a description of the technique itself.

Ontology Alignment Systems

Ontology alignment is intended for use in an automated fashion for two primary reasons: first, it's a time-consuming, tedious, and occasionally difficult task, and, second, its true value is revealed when it is integrated into processes that exchange information automatically. This has resulted over the past few years in the

emergence of multiple software tools that have been developed by diverse research groups and well-established international organizations, primarily associated with the academic community. The tools, designed to automatically identify the correspondences that may exist between entities of different ontologies, are called *ontology alignment systems*.

Through the development of these tools, a considerable number of ontology alignment systems have become available. Each one of these systems offers a unique set of advantages, disadvantages, and performance characteristics. Table 2 lists the main ontology alignment systems that are currently available.

An ontology alignment system accepts one (or more) ontologies as input, and provides, as output, a set of correspondences between their elements. This set of correspondences is referred to as *alignment*. The quality of a particular alignment depends on the correctness and completeness of the correspondences it has found. An alignment system is typically based on several of the latest alignment techniques in conjunction with its own methods with the aim of obtaining the most precise and complete alignment possible.

FUTURE TRENDS

At present, there are several ontology alignment systems capable of identifying, with acceptable efficiency, semantic correspondences that may exist

Table 2. Ontology alignment systems

Name	Developed by	References
Anchor-PROMPT	Stanford University (USA)	Noy & Musen, 2003
Chimaera	Stanford University (USA)	McGuinness, Fikes, Rice & Wilder, 2000
CMS	School of Electronics and Computer Science & Advanced Knowledge Technologies group (University of Southampton), Hewlett Packard Laboratories (UK)	CMS, 2006, Kalfoglou & Hu, 2005
COMA++/ COMA	University of Leipzig (Germany)	COMA, 2006, Aumueller, Do, Massmann & Rahm, 2005, Massmann, Engmann & Rahm, 2006
CtxMatch	University of Trento (Italy)	Zanobini, 2004
Blue	University of Washington (USA)	Doan, Madhavan, Domingos & Halevy, 2002, Doan, Madhavan, Domingos & Halevy, 2004
Falcon-AO	Southeast University (China)	Jian, Hu, Cheng & Qu, 2005, Hu, Jian, Qu & Wang, 2005, Hu, Zhao & Qu, 2006, Hu, Cheng, Zheng, Zhong & Qu, 2006
FOAM [APFEL, NOM, QOM]	University of Karlsruhe (Germany)	Ehrig & Staab, 2004, Ehrig & Sure, 2005, Ehrig, Staab & Sure, 2005
HCONE-merge	University of Aegean (Greece)	Kotis, Vouros & Padilla, 2004, Kotis, Vouros & Stergiou, 2005, Vouros & Kotis, 2005
H-Match	University of Milan (Italy)	Castano, Ferrara & Montanelli, 2003
LOM	Teknowledge Corporation (Palo Alto, USA)	Li, 2004
MAFRA	Instituto Politecnico do Porto (Portugal)	Maedche, Motik, Silva & Volz, 2002
MapOnto	University of Toronto (Canada), University of Rutgers (USA)	An, Borgida & Mylopoulos, 2005
MetaQuerier	University of Illinois (USA)	Chang, He & Zhang, 2004, Chang. He & Zhang, 2005
MoA	Electronics and Telecommunications Research Institute (Korea)	Jaehong et al., 2005
OLA	INRIA Rhône-Alpes (France), University of Montreal (Canada)	Euzenat, Loup, Touzani & Valtchev, 2004, Euzenat & Valtchev, 2004, Euzenat, Guérin & Valtchev, 2005
OntoBuilder	Technion Israel Institute of Technology (Israel)	Gal, Modica & Jamil, 2004
OntoMerge	Yale University (USA), University of Oregon (USA)	Dou, McDermott & Qi, 2002
Rondo	University of Leipzig (Germany), Microsoft Research (USA)	Melnik, Rahm & Bernstein, 2003
S-Match	University of Trento, Italy	Giunchiglia, Shvaiko & Yatskevich, 2004
SAMBO	University of Linköpings (Sweden)	Lambrix & Tan, 2006

O

between entities associated with different ontologies. However, the true potential of ontology alignment will be realized when this methodology is integrated in processes that require that information between different systems be exchanged fully automatically. This would be achievable when ontology alignment systems become sufficiently powerful to resolve, in real time and with minimal error, alignment problems in specific domains.

Once these issues are successfully addressed, it will become possible to attain an appropriate level of interoperability between heterogeneous systems that were previously not exploited jointly, thereby representing a high water mark in the field of information and communications technologies. Multiple systems of different characteristics and origins would thus be able to communicate with each other, making it possible to reveal new knowledge that could have previously remained uncovered in disjointed information systems. This would potentially provide human users with a wide range of automated intelligent systems and services capable of interrelating with each other without external assistance, which in turn would considerably facilitate one of the most challenging tasks: the automatic, efficient, and reliable exploitation of large quantities of information.

CONCLUSION

In some applications, the use of a single ontology to fully describe an entire domain is generally not an adequate solution, and it normally becomes necessary to use different ontologies. In such cases, the need arises to find relationships between the elements of the different ontologies, a process known as ontology alignment.

Automation of the ontology alignment process can be reasonably achieved, which is precisely why this process is especially useful in environments or applications that require the automatic interoperability between systems. Currently, there are numerous ontology alignment systems available, and most of these are the result of academic or basic research. These systems can be viewed as software tools capable of finding correspondences or relationships that may exist between the elements of different ontologies. These tools can provide rather remarkable results, especially when taking into account the fact that they essentially remain

works in progress, still in the initial development or testing phases.

In the future, it is expected that ontology alignment systems will reach acceptable levels of robustness, efficiency, and reliability, which would make it possible to apply these systems to processes that automatically exchange data between different systems that individually utilize different ontologies. These automated interactions between systems would not only reduce user intervention but would also automate many time-consuming, complex, and computationally costly tasks that are currently either performed manually or not at all.

ACKNOWLEDGMENT

REFERENCES

An, Y., Borgida, A., & Mylopoulos, J. (2005). Constructing Complex Semantic Mappings between XML Data and Ontologies. *Proceedings of ISWC'05*.

Aumueller, D., Do, H.H., Massmann, S., & Rahm, E. (2005). Schema and ontology matching with COMA++. *SIGMOD Conference*.

Castano, S., Ferrara, A., & Montanelli, S. (2003). H-MATCH: an algorithm for dynamically matching ontologies in peer-based systems. *Proceedings of the First Workshop on Semantic Web and Databases (SWDB-03)*, VLDB 03, Berlin, Germany.

Chang, C., He, B., & Zhang, Z. (2004). MetaQuerier over the Deep Web: Shallow Integration across Holistic Sources. *Proceedings of the VLDB Workshop on*

Information Integration on the Web (VLDB-IIWeb'04), Toronto, Canada.

Chang, C., He, B., & Zhang Z. (2005). Towards Large Scale Integration: Building a MetaQuerier over Databases on the Web. *Proceedings of the Second Conference on Innovative Data Systems Research (CIDR 2005)*, Asilomar, California.

COMA Website (2006). URL: http://dbs.uni-leipzig.de/en/Research/coma.html/

Crosi Mapping System Website (2006). URL: http://www.aktors.org/crosi/deliverables/summary/cms.html/

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2002). Learning to map between ontologies on the semantic web. *Proceedings of the World-Wide Web Conference*, Hawai, USA.

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2004). Ontology Matching: A Machine Learning Approach. Staab, S. & Studer, R. (eds.). *Handbook on Ontologies in Information Systems*, Springer-Velag, 397-416.

Dou, D., McDermott, D., & Qi, P. (2002). Ontology translation by ontology merging and automated reasoning. *Proceedings of the EKAW 2002 Workshop on Ontologies for Multi-Agent Systems*. Sigüenza, Spain.

Ehrig, M., & Staab, S. (2004). QOM - Quick Ontology Mapping. *Proceedings of the Third International Semantic Web Conference*, LNCS 3298, 683-697. Springer, Hiroshima, Japan.

Ehrig, M., & Sure, Y. (2005). FOAM - Framework for Ontology Alignment and Mapping - Results of the Ontology Alignment Evaluation Initiative. *Proceedings of the Workshop on Integrating Ontologies*, 156, 72-76.

Ehrig, M., Staab, S., & Sure, Y. (2005). Bootstrapping Ontology Alignment Methods with APFEL**.** *Proceedings of the 4th International Semantic Web Conference, ISWC 2005,* LNCS 3729, 186-200. Springer.

Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., De Bo, J., Dieng, R., Ehrig, R., et al. (2004). State of the art on ontology alignment. *Deliverable D2.2.3 v1.2*. Knowledge Web. URL: http://knowledgeweb.semanticweb.org/

Euzenat, J., & Valtchev, P. (2004). Similarity-based ontology alignment in OWL-Lite. *Proceedings of 16th european conference on artificial intelligence (ECAI)*, 333-337. Amsterdam, Holland.

Euzenat, J., Loup, D., Touzani, M., & Valtchev, P. (2004). Ontology alignment with OLA. *Proceedings of 3rd ISWC2004 workshop on Evaluation of Ontology-based tools (EON)*, 59-68, Hiroshima, Japan.

Euzenat, J., Guérin, P., & Valtchev, P. (2005). OLA in the OAEI 2005 alignment contest. *Proceedings K-Cap 2005 workshop on Integrating ontology,* 97-102, Banff, Canada.

Gal, A., Modica, G. A., & Jamil, H. M. (2004). OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources. *Proceedings of the ICDE 2004*.

Giunchiglia, F., Shvaiko, P., & Yatskevich, M. (2004). S-Match: An Algorithm and an Implementation of Semantic Matching. *Proceedings of ESWS'04*.

Gruber, T. R. A translation approach to portable ontology specification. (1993). *Knowledge Acquisition*, 5(2), 199-200.

Hu, W., Jian, N., Qu, Y., & Wang, Y. (2005). GMO: A graph matching for ontologies. *Proceedings of the K-CAP workshop on Integrating Ontologies*, 41-48.

Hu, W., Zhao, Y., & Qu, Y. (2006). Partition-based block matching of large class hierarchies. *Proceedings of the 1st Asian Semantic Web Conference (ASWC'06)*, 72-83.

Hu, W., Cheng, G., Zheng, D., Zhong, X., & Qu, Y. (2006). The Results of Falcon-AO in the OAEI 2006 Campaign. *ISWC Ontology matching workshop*. Athens, USA.

Jaehong, K., Jang, M., Young-Guk, H., Joo-Chan, S. & Jo, S. (2005). MoA: OWL ontology merging and alignment tool for the semantic web. *Lecture notes in Computer Science*, 3533/2005, 722-731, Springer.

Jian, N., Hu, W., Cheng, G., & Qu, Y. (2005). Falcon-AO: Aligning Ontologies with Falcon. *Proceedings of K-Cap 2005 Workshop on Integrating Ontologies*, 85-91, Banff, Canada.

Kalfoglou, Y., & Hu, B. (2005). CMS: CROSI Mapping System - Results of the 2005 Ontology Alignment

O

Contest. *Proceedings of K-Cap '05 Integrating Ontologies workshop*, 77-85, Banff, Canada.

Kotis, K., Vouros, G. A., & Padilla, J. (2004). HCOME: tool-supported methodology for collaboratively devising living ontologies. *Semantic Web and Databases. Second International Workshop, SWDB*. Toronto, Canada.

Kotis, K., Vouros, G., & Stergiou, K. (2005). Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach. *Elsevier's Journal of Web Semantics (JWS)*, 4:1, 60-79.

Lambrix, P., & Tan, H. (2006). SAMBO - A System for Aligning and Merging Biomedical Ontologies. *Journal of Web Semantics, Special issue on Semantic Web for the Life Sciences*, 4(3), 196-206.

Li, J. (2004). LOM: A Lexicon-based Ontology Mapping Tool. *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS. '04)*.

Maedche, A., Motik, B., Silva, N., & Volz, R. (2002). MAFRA - A Mapping Framework for Distributed Ontologies. *Proceedings of 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW)*. Sigüenza, Spain.

Martínez, M. (2007). Analysis and comparative study of ontology alignment systems, and development of an ontology alignment system optimized for aligning medical ontologies. Pazos, A., Vázquez, J.M. (dirs.). University of A Coruña. Final project.

Massmann, S., Engmann, D., & Rahm, E. (2006). COMA++: Results for the Ontology Alignment Contest OAEI 2006. *International Workshop on Ontology Matching (5th ISWC-2006)*, Athens, Georgia, USA.

Melnik, S., Rahm, E., & Bernstein, P. A. (2003). Rondo: A Programming Platform for Model Management. *Proceedings of ACM SIGMOD 2003*, San Diego, USA.

McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000). An environment for merging and testing large ontologies. *Proceedings of 7th Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR2000)*. Colorado, USA.

Noy, F. N., & Musen, A. M. (2003). The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. *International Journal of Human-Computer Studies*, 59/6, 983-1024.

Vouros, G., & Kotis, K. (2005). Extending HCONE-merge by approximating the intended interpretations of concepts iteratively. *2nd European Semantic Web Conference*, Heraklion, Creta, Greece.

WordNet, 2007. Cognitive Science Laboratory. Princeton University. URL: http://wordnet.princeton.edu/

Zanobini, S. (2004). *Improving ctxmatch by means of grammatical and ontological knowledge - in order to handle attributes*. Technical Report 554, Department of Information and Communication Technology, University of Trento, Italy.

KEY TERMS

Class: A set that contain individuals which share certain characteristics. The word *concept* is sometimes used in place of class. Classes are a concrete representation of concepts.

Individual: A object in the domain that we are interested in. Individuals are also known as *instances of classes*.

Interoperability: A state or situation through which heterogeneous systems can exchange data and/or processes.

Mapping: A correspondence found during the process of ontology alignment.

Ontology: A formal and explicit specification of a shared conceptualization.

Ontology Alignment: A process that consists of finding the semantic relationships that may exist between different elements in different ontologies.

Ontology Alignment System: A software tool capable of conducting the alignment of ontologies in an automated fashion.

Ontology Mapping: See ontology alignment.

Ontology Matching: See ontology alignment.

Relation: A link between individuals. In the field of ontologies, relations are also known as *properties*.

Ontology Alignment Techniques

Marcos Martínez Romero
University of A Coruña, Spain

José Manuel Vázquez Naya
University of A Coruña, Spain

Javier Pereira Loureiro
University of A Coruña, Spain

Norberto Ezquerra
Georgia Institute of Technology, USA

INTRODUCTION

Sometimes the use of a single ontology is not sufficient to cover different vocabularies for the same domain, and it becomes necessary to use several ontologies in order to encompass the entire domain knowledge and its various representations. Disciplines where this occurs include medical science and biology, as well as many of its associated subfields such as genetics, epidemiology, etc. This may be due to a domain's complexity, expansiveness, and/or different perspectives of the same domain on the part of different groups of users. In such cases, it is essential to find relationships that may exist between the elements of a specific domain's different ontologies, a process known as ***ontology alignment***.

There are several methods for identifying the relationships or correspondences between elements associated with different ontologies, and collectively these methods are called ***ontology alignment techniques***. Many of these techniques stem from other fields of study (e.g., matching techniques in discrete mathematics) while others have been specifically designed for this purpose. The key to successfully aligning ontologies is based on the appropriate selection and implementation of a set of those ontology alignment techniques best suited for a particular alignment problem.

Ontology alignment is a complex, tedious, and time-consuming task, especially when working with ontologies of considerable size (containing, for instance, thousands of elements or more) and which have complex relationships between the elements (for example, a particular problem domain in medicine). Furthermore, the true potential of ontology alignment is realized when different information-exchange processes are integrated automatically, thereby providing the framework for reaching a suitable level of efficient interoperability between heterogeneous systems. The importance of automatically aligning ontologies has therefore been a topic of major interest in recent years, and recently there has been a surge in a variety of software tools dedicated to aligning ontologies in either a fully or partially automated fashion. Some of these tools —generally referred to as ***ontology alignment systems***— have been the result of well known and respected research centers, including Stanford University and Hewlett Packard Laboratories, for instance. In Shvaiko & Euzenat, 2007, updated information is given regarding the currently available ontology alignment systems.

Each ontology alignment system combines different alignment approaches along with its own techniques, such that correspondences between the different ontologies can be detected in the most complete, precise, and efficient manner. Since each system is based on its own approximation techniques, different systems yield different results, and therefore the quality of the results can vary among systems. Most of the alignment systems are oriented to solving problems of a general nature, since ontologies associated with a single domain share certain characteristics that set them apart from ontologies associated with other domains. Recently, some systems have emerged that are designed to align ontologies in a specific domain. An example is the SAMBO alignment system (Lambrix, 2006) in the biomedical domain. These and other domain-specific systems can produce excellent results (when used for the domains for which they were designed), but are generally not useful when applied to other domains.

This article presents a classification of the most commonly used, recently developed alignment techniques, supported by simple examples to illustrate the specific techniques underlying different systems. Future directions in ontology alignment are also examined.

BACKGROUND

The key to ontology alignment is to find those entities in one ontology that may correspond to other entities in another ontology. Basically, this can be viewed as finding a similarity measure between elements (or so-called entities) associated with different ontologies, and subsequently selecting the set of correspondences that produce the strongest measures of similarity. There are, however, different ways to compute similarity measures; there are various studies dedicated to the classification of these techniques (Rahm & Bernstein, 2001, Euzenat & Valtchev, 2004, Euzenat et al., 2004, Shvaiko & Euzenat, 2005).

Following these classification schemes (especially those undertaken by Euzenat and Valtchev (Euzenat & Valtchev, 2004) and based on Euzenat et al., 2004), the next section will introduce an abbreviated classification of those ontology alignment techniques that are most commonly utilized by current ontology alignment systems. This condensed classification is centered on the type of element being manipulated by the alignment technique, and complements the taxonomy proposed by Rahm and Bernstein (Rahm & Bernstein, 2001), and —for the purpose of clarity and brevity—summarizes only those alignment techniques that compare on an individual basis a single element in one ontology with another element associated with another ontology (known as local alignment techniques, as in Euzenat et al., 2004).

ONTOLOGY ALIGNMENT TECHNIQUES

Ontology alignment techniques can be classified according to the following (please refer to Figure 1):

1. **Terminological techniques.** These calculate the similarity between text strings and describe several elements in the ontologies (names, labels, and/or comments). There are two types of terminological

techniques: those based on text strings and those based on the language.

1.1. Terminological techniques based on text strings. These are based on the idea of comparing the structure in text strings, which are viewed as sequences of characters. These techniques consider that the similarity between two terms increases when the similarity between their corresponding text strings also increases, but without considering the underlying semantics in the terms. In this manner, the application of a technique of this type to the terms *Apple* and *Apples* would yield a relatively high measure of similarity, whereas the application of the same technique to the terms *Apple* and *Orange* would yield a lower degree of similarity (or a lower similarity measure), since in the second case the text strings are quite different. The isolated use of these techniques is usually not recommended, since it is preferable to use them in conjunction to other, more powerful alignment techniques; these can be easily illustrated with the following example: it would be erroneous to conclude that the terms *Cream* and *Scream* are highly similar (although their meanings are very different), or that the terms *Student* and *Pupil* are very distinct or dissimilar (although the semantic concepts are generally the same). Some examples of terminological techniques based on text strings are the distance measure proposed by *Hamming* (Hamming, 1950), which counts the number of different characters in two different text strings; the distance measure suggested by *Levenshtein* (Levenshtein, 1966), which examines the minimum number of operations (insertions, deletions and/or substitutions) that are necessary to transform one text string into another; and the distance measure *Jaro* (Jaro, 1989), which analyzes the number and order of two common characters in two text strings.

1.2. Terminological techniques based on language. These techniques are more complex but more reliable than those previously discussed, and do not treat terms as simple sequences of characters that are independent of one another. Rather, these techniques view terms as groups of elements with meaning (lexima and morphema, i.e., prefixes and suffixes). The main objective of these techniques is to discover the similarity that may exist between terms associated with one concept, although the relationships can be formed by strings of characters that are very different. In other words, these techniques attempt to obviate the different termino-

logical variations that can affect terms that are being mutually compared. These techniques, in turn, can be classified according to whether intrinsic and extrinsic approaches:

1.2.1. Intrinsic techniques. These are oriented toward detecting the similarity between terms that have undergone morphological and syntactical variations (e.g, *Mean of transport, Mean of transportation, Transportation mean*), as in Porter Stemming Algorithm (Porter, 1980).

1.2.2. Extrinsic techniques. These consist of utilizing external linguistic resources, such as dictionaries and thesaurus, in order to find the similarity between lexical variations in the same term (e.g, *Mean of transport* and *Vehicle*). External techniques consider the fact that there usually is an equivalence relationship between synonyms, and a subsuming relationship between hyponyms. In this manner, an alignment system based on extrinsic terminological techniques would presumably be capable of detecting, for instance, an equivalence relationship between the terms *Leukocyte* and *White blood cell* (since they are synonymous) and a subsumed relationship between *Moycyte* and *Cell*

(since *Myocyte* is a type of *Cell*). Some of the external linguistic resources most commonly used by such alignment systems currently in use include WordNet (WordNet, 2007), as an English-language resource, or UMLS (National Library of Medicine, 2007) in the medical domain. Other extrinsic techniques that are in use include *multilingual techniques*, dedicated to finding relationships between terms written in different languages (such as the Spanish word *célula* and its English counterpart, *cell*) and using multilingual dictionaries such as EuroWordNet (Vossen, 1997).

2. **Structural Techniques.** In addition to comparing text strings that describe the entities in each ontology, it is frequently useful to compare the internal structure of the entities themselves, or the relationships that each entity may maintain with other entities (external structure comparison).

2.1. Internal structure comparison techniques. These techniques compare internal characteristics of the entities, such as the rank, cardinality, transitivity, and/or symmetry of its properties (attributes and relationships). For instance, if in one ontology *A* there is an entity *Per-*

Figure 1. Ontology alignment techniques
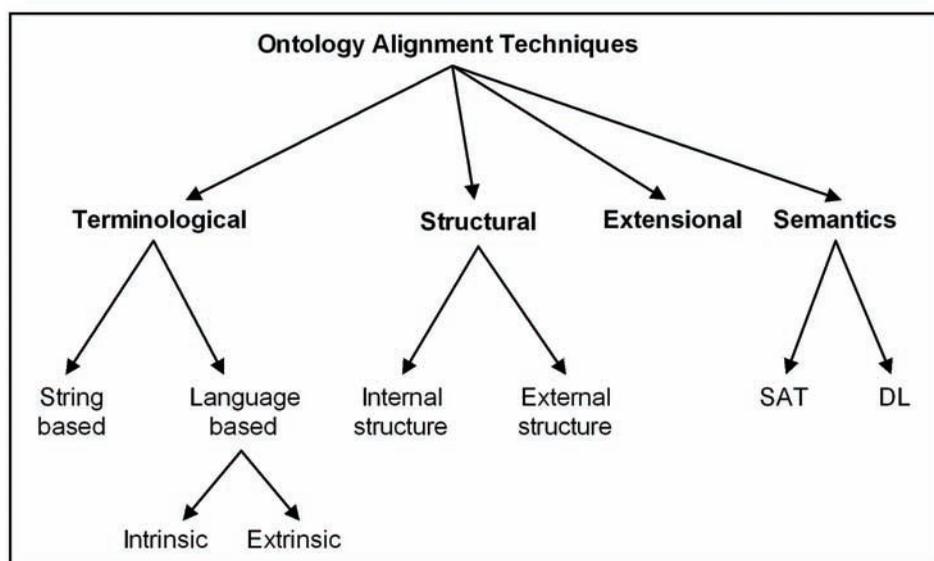
son with three attributes (*birth_date* of type *date*; *name* of type *string*, and *weight* of type *int*), and in another ontology *B* there is an entity *Human_being* with two attributes (*date_of_birth* of type *date*; and *first_name* of type *string*), a technique of this type might conclude that there is certain similarity between these two entities, since the types of two of the attributes coincide. In this concrete case the technique's conclusion would have been correct: *Person* and *Human_Being* refer to the same concept in the real world. However, it is easy to find cases in which the technique would produce erroneous results. For instance, if the entity in ontology *B* were *Car* with three attributes (*registration_date* of type *date*; *color* of type *string*; and *weight* of type *int)*, a comparison of internal structure might suggest that the entities *Person* and *Car* were similar, since the ranks of the three attributes coincide although in reality they are entities associated with very different semantics. Consequently, given that it is frequently possible to find multiples entities in an ontology that represent similar internal characteristics, these techniques tend to be used in conjunction with other techniques (such as terminological techniques). It is probably wise to utilize a method to compare the internal structure during the initial alignment stages, in order to filter pairs of entities that could be related, and subsequently apply other techniques before finally deciding on the overall level of similarity.

2.2. External structure comparison techniques.
These techniques compute the similarity that may exist between entities by considering the position that the entities in question occupy within their respective ontologies. The underlying principle is that, if two entities are similar, then there is likely to be some similarity with their adjacent (or neighboring) entities. These techniques tend to treat ontologies as graphs in which each node is a vertex in the ontology and each edge is a relationship between vertices; algorithms that are especially designed to work with graphs are used to find the relationships between elements in the ontologies. As a matter of fact, this problem is equivalent to that or solving a graph homomorphism (Garey, 1979). One of the better known techniques for making the external-structure comparison is the one used by the Anchor-PROMPT ontology alignment system (Noy & Musen, 2000), which is based on the idea that if two pairs of entities in the source ontologies are similar and there are connected paths linking them, then the elements in those paths are also similar.

3. **Extensional techniques.** These extensional (or extensible) techniques compare the extension or length of the classes of ontologies: in other words, their instantiations or examples. This is useful when the information about the entities to be compared is limited but there is additional data or information about their examples; alternately, they are useful as a means of supporting other alignment techniques in order to detect erroneous or misleading correspondences. For instance, if an ontology contains a class denoted as *Human_being* with two instances, *John* and *Mary*, and the other ontology contains a class labeled *Person* with the same instances (*John* and *Mary*), then it could be inferred, by comparing all the instances of the ontologies, that the classes are similar.

4. **Semantic techniques.** These types of techniques attempt to align the elements in the ontologies according to their semantic interpretation. The general approach is based on deductive methods that draw from theoretical models that provide a justification for the results that are obtained. Some examples include the Propositional SATisfiability (SAT) and techniques based on Description Logics (DL).

4.1. SAT techniques: the application of SAT techniques to the ontology alignment problem consists of translating the information associated with pairs of terms between which a mathematical or formulaic relationship could exist. The relationship would be of the form *Axioms*→*rel*(*element*$_1$, *element*$_2$), where *element*$_1$ and *element*$_2$ are the entities in the ontologies that are being examined to determine if there is a semantic relationship between them, and *rel* is the relationship that exists between the entities. Subsequently, the validity of the relationship (the aforementioned formula) is evaluated. The advantage of using SAT techniques is that it supports an exhaustive analysis of all the possible correspondences as well as the possibility of selecting only the major correspondences.

4.2. Techniques based on DL: the expressivity of propositional language used by SAT techniques is limited, as they are unable to work with certain types of predicates. However, Description Logics provides the necessary expressivity to code alignment problems as propositional validity problems with greater flexibility. For instance, if an ontology contains the classes *City*,

Worker and *Industrial_city*, as a *City* with more than 600,000 Workers, and another ontology contains the classes *Big_town, Inhabitant* and *Crowded_big_town*, as a *Big_town* with more than 500,000 *Inhabitants* and it is established that all *Workers* are *Inhabitants* and that *City* is equivalent to *Big_town*, then a DL-based technique could deduce that an *Industrial_city* is a *Crowded_big_town*.

FUTURE TRENDS

Current ontology alignment systems take as input two ontologies and, once the alignment process is executed, yield as output a set of correspondences between their elements. Using up-to-date alignment techniques, this process is still very time consuming and computationally expensive especially in those cases where the input ontologies are large. This may not present a challenge in cases where the same ontologies are always used, since in such cases it would only be necessary to perform the alignment once, and subsequently the correspondences that have been revealed could be reutilized.

However, there are applications or contexts where it becomes necessary to instantly identify which entity in ontology *A* corresponds with an entity in ontology *B*, without previously "knowing" the ontologies. In these cases, current ontology alignment techniques are limited, as is the case with the Semantic Web or the integration of information from different sources that were mutually "unknown" to each other. In these types of problems, it is more important to reduce the computational time that is necessary to carry out the alignment, although the quality of the alignment could be somewhat affected. As a result, it is very probable that in the next few years the field of ontology alignment will see a major thrust being placed on exploring techniques capable of finding correspondences in an increasingly shortened amount of time.

It is also expected that new techniques will emerge that will allow the consultation or usage of external linguistic resources in a more efficient and powerful manner than is now possible. The utilization of external resources is essential in alignment problems associated with specific domains, although current approaches are not capable of achieving optimal usage of these types of resources, thereby wasting a significant amount of potentially useful information.

CONCLUSION

Ontology alignment is an important aspect of practically any domain or application area where it is necessary to use an ontology. There are various approaches to finding semantic correspondences that may exist between elements of different ontologies, known as ontology alignment techniques. This paper has presented a condensed classification of those ontology alignment techniques that are most commonly used today.

Clearly, not all alignment techniques are equally applicable to any problem. For instance, it is not useful to apply an extensional technique to ontologies that have no instances. Consequently, a number of factors ought to be considered when selecting among different alignment techniques for application to a particular problem. Among these are the domain to which the ontologies belong, the language in which the ontologies are expressed, the number and type of elements contained in the ontologies, etc. And, although a particular technique may be applicable to a specific alignment problem, there is also the question of errors. As a result, it should be stressed that aligning two ontologies is not simply the application of an alignment technique in an isolated manner: rather, the goal is mainly to find the appropriate combination of alignment techniques to be applied, such that the strengths of one technique can compensate another technique's weaknesses and limitations, with the overarching objective of uncovering an optimal set of correspondences between the ontologies of interest.

ACKNOWLEDGMENT

REFERENCES

Euzenat, J., Loup, D., Touzani, M., & Valtchev, P. (2004). Ontology alignment with OLA. *Proceedings of 3rd ISWC2004 workshop on Evaluation of Ontology-based tools (EON)*. 59-68. Hiroshima, Japan.

Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., De Bo, J., Dieng, R., Ehrig, R., et al. (2004). *State of the art on ontology alignment*. Deliverable D2.2.3 v1.2. Knowledge Web. URL: http://knowledgeweb.semanticweb.org/

Euzenat, J., & Valtchev, P. (2004). Similarity-based ontology alignment in OWL-Lite. *Proceedings of 16th european conference on artificial intelligence (ECAI)*, 333-337. Amsterdam, Holland.

Garey, M., & Johnson, D. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman & Co.

Gruber, T. R. A translation approach to portable ontology specification. (1993). *Knowledge Acquisition*, 5(2), 199-200.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 26(2):147-160.

Jaro, M. A. (1989). Advances in record linking methodology as applied to the 1985 census of Tampa Florida. *Journal of the American Statistical Society*, 64:1183-1210.

Lambrix, P., Tan, H. (2006). SAMBO - A System for Aligning and Merging Biomedical Ontologies. *Journal of Web Semantics, Special issue on Semantic Web for the Life Sciences*, 4(3), 196-206.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Noy, F. N., & Musen, A. M. (2000). Anchor-PROMPT: Using non-local context for semantic matching. *Proceedings of the Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*. Seattle, USA.

Shvaiko, P., Euzenat, J. (2007). Ontology Matching Web. URL: http://www.ontologymatching.org

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program*, 14(3): 130-137.

Rahm, E., & Bernstein, P. (2001). A Survey of Approaches to Automatic Schema Matching. *VLDB Journal,* 10(4), 334-350.

Shvaiko, P., & Euzenat, J. (2005). A Survey of Schema-based Matching Approaches. *Journal on Data Semantics (JoDS)*, IV, LNCS 3730,146-171.

National Library of Medicine (NLM), (2007). *Unified Medical Language System*. URL: http://umlsinfo.nlm.nih.gov/

Vossen, P. (1997). EuroWordNet: a multilingual database for information retrieval. *Third DELOS workshop – Cross-Language Information Retrieval. European Research Consortium for Informatics and Mathematics*, 85-94, Zurich.

WordNet. (2007). Cognitive Science Laboratory. Princeton University. URL: http://wordnet.princeton.edu

KEY TERMS

Domain: Specific areas of interest (e.g., artworks by Picasso) or of knowledge (e.g., medicine, physics, etc.).

Ontology: A formal and explicit specification of a shared conceptualization.

Ontology Alignment: A process that consists of finding the semantic relationships that may exist between different elements in different ontologies.

Ontology Alignment System: A software tool capable of conducting the alignment of ontologies in an automated fashion.

Ontology Alignment Technique: Method used to identify the semantic correspondences that may exist between the elements of different ontologies.

Ontology Entity: An ontology entity represents a conceptual element of the domain of discourse.

Thesaurus. Networked collection of controlled vocabulary terms.

Optimization of the Acoustic Systems

V. Romero-García
Polytechnic University of Valencia, Spain

E. Fuster-Garcia
Polytechnic University of Valencia, Spain

J. V. Sánchez-Pérez
Polytechnic University of Valencia, Spain

L. M. Garcia-Raffi
Polytechnic University of Valencia, Spain

X. Blasco
Polytechnic University of Valencia, Spain

J. M. Herrero
Polytechnic University of Valencia, Spain

J. Sanchis
Polytechnic University of Valencia, Spain

INTRODUCTION

A genetic algorithm is a global search method based on a simile of the natural evolution. Genetic Algorithms have demonstrated good performance for difficult problems where the function to minimize is complicated. In this work we applied this optimization method to improve the acoustical properties of the Sonic Crystal (Martínez-Sala et Al.,1995) (Kushwaha et al., 1994), a kind of structures used in acoustics.

In the last few years the propagation of the acoustic waves in heterogeneous materials whose acoustic properties vary periodically in space have attracted considerable interest. The so-called Sonic Crystals are the typical example of this kind of materials in the range of the acoustic frequencies. These systems are defined as periodic structures with strong modulation of the elastic constants between the scatterers and the surrounding material.

Recently, the strategy to enhance Sonic Crystals properties has been based on the use of scatterers with acoustical properties added. The use of local resonators (Liu et al., 2000) or Helmholtz resonators (Hu et al., 2005) as scatterers have produced very good results Some authors also have built new structures with scatterers made up of porous material improving the attenu-

ation capability of the Sonic Crystals (Umnova et al., 2006). However, the use of Sonic Crystals as outdoor acoustic barriers requires scatterers made up of robust and long-lasting materials. This is the reason why it seems interesting to analyze the possibility of optimizing the attenuation capability of Sonic Crystals made with rigid scatterers like wood, PVC or aluminium. The creation of vacancies in a Sonic Crystals improves the attenuation capability of the Sonic Crystals (Caballero et al., 2001). However, it does not exist any generic rule about the creation of vacancies in a Sonic Crystals. In fact, similar structures can produce very different acoustic fields behind of them.

Because of the complexity of mathematical functions involved in Sonic Crystals calculus, Genetic Algortihm turns up as a tool specially indicated for this kind of problems (Hakanson et al., 2004) (Romero-García et al., 2006). This procedure can work together with the Multiple Scattering theory which is a self-consistent method for calculating the acoustic pressure including all orders of scattering (Chen & Ye, 2001). Given a starting Sonic Crystals, the Genetic Algorithm generates quasi ordered structures offspring by means of the creation of vacancies that are classified in terms of a cost function based on the pressure values at a specific point. The sound scattered pressure by every

structure analyzed by Genetic Algorithm is performed by a two-dimensional (2D) Multiple Scattering theory. In the present work, it is shown an improvement of the Genetic Algorithm based on Parallel implementation and as a consequence, new and better results are obtained to design Quasi Ordered Structures made with rigid cylinders that attenuate sound in a predetermined band of frequencies.

SONIC CRYSTALS

Sonic Crystals are arrays of scatterers placed periodically in space whose physical properties are different to the surrounding material. In the low frequency range, Sonic Crystals behave as an homogeneous medium with an acoustic impedance greater than that of the air. Then Sonic Crystals can work as refractive devices. Moreover, Sonic Crystals present band gaps, i.e., ranges of sound frequencies where the sound propagation inside the crystal is forbidden. The presence of these band gaps is explained by the well-known Bragg's law. The reflections inside the crystal, and consequently the position of the gaps depend on the lattice constant, i.e., on the geometry of the Sonic Crystals. The existence, in periodic media, of an absolute band gap where the propagation of sound is forbidden for every incidence direction, can have a profound impact on several scientific and technological disciplines, for example, in the design of acoustic filters or acoustic barriers.

Some studies have showed that there are three important parameters for the spectral gap creation (Economolu & Sigalas, 1994). One is the density ratio $y = \rho_s/\rho_h$ between the scattering material and the host material densities. The second one is the filling factor, $ff = Vs/V$, that shows the volume occupied by the scattering material respect to the total volume. The last parameter is the topology used to design the Sonic Crystals. It was demonstrated that the density ratio plays an important role in the gap creation: Sonic Crystals built with scatterers of high density embedded in a host material of low density are better to create the spectral gap than another kind of configurations. Moreover the optimum value of the filling factor, ff, to the gap creation has been ranged between 10% and 50%. In this work we use a Sonic Crystals built by aluminium cylinders of 2 cm of radius as scatterers embedded in air (Network topology). Due to the fact that those structures present a high density ratio, and the

maximum filling factor is $ff = 0,36$, we ensure that our structure is well designed to the gap creation. Now we want to find the best filling factor and space distribution of scatterers that present the best acoustical properties. Genetic Algorithm together with the MST is a good procedure to achieve our objective.

COST FUNCTION AND CHROMOSOME DESCRIPTION

The mechanism used by Genetic Algorithm in this work is the creation of vacancies in the starting Sonic Crystals. Fig. 1 shows the starting Sonic Crystals and a Quasi Ordered Structures offspring generated by Genetic Algorithm by means of the creation of vacancies. Using this procedure we can vary the filling factor and, at the same time, evaluate different spaces of configuration. Each Quasi Ordered Structures will be considered as an individual. The chromosome that represents each Quasi Ordered Structures, is a real vector with values in [0*;* 1] range. Each coordinate represents the existence or not of a cylinder at a specific position of the scatterer (beginning with the cylinder a the left top corner of the Sonic Crystals and following by columns until right bottom corner, see starting Sonic Crystals at figure 1). Values in [0*;* 0*:*5[means there is a vacancy, in opposition values in [0*:*5*;* 1] means there is a cylinder. In this work we are interested in maximizing the sound attenuation for a predetermined range of frequencies not dependent on the lattice constant, at a point located behind the crystal.

The acoustic attenuation in a point (x, y) and for a incidence frequency v is:

$$\text{Atenuación (dB)} = 20\log\left(\frac{1}{|P_{interfered}(x, y, Xcil, Ycil, v, r_1)|}\right)$$

where the interfered pressure is determined by the MST. This pressure depends on the position and on the radius of the scatterers and the incidence frequency. In the equation (1) we can see that for a point (x, y), a value of incidence frequency v and a value of cylinder radius r_1, it is possible to find a configuration of cylinders that minimize the $P_{interferred}$, that means, maximize the acoustic attenuation.

If we are interested in maximizing the sound attenuation in a predetermined range of frequencies at

Figure 1. Starting sonic crystals and a possible quasi ordered structures offspring
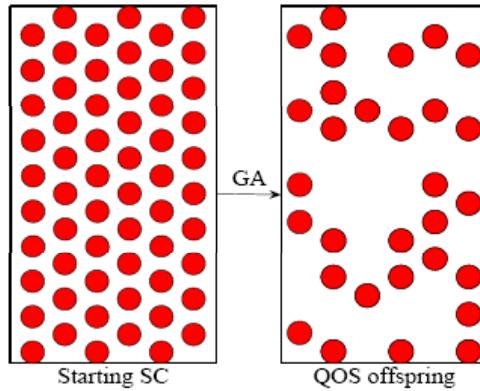
a point of coordinates (*x, y*) we have to define a new function that we have to minimize in order to achieve the maximum acoustic attenuation. To do that, we define our cost function based on the MST

$$J = \frac{\sum_{v=1}^{N_v} p_v}{N_v} + \sqrt{\frac{\sum_{v=1}^{N_v} (\bar{p} - p_v)}{N_v^2}}$$

where

$$\bar{p} = \sum_{j=1}^{N_v} \frac{|p_j|}{N}$$

represents the mean pressure in the range of frequencies $[v_1; v_N]$ and N represents the number of frequencies considered in this range. In our case, we use $N = 13$. The second term in equation (2) represents the mean deviation. The variable under study is $\mathbf{x}=(Xcyl, Ycyl)$ a vector that contains the information about the space configuration of the Quasi Ordered Structures.

PARALLEL GENETIC ALGORITHM

A Genetic Algorithm is an optimization technique that looks for the solution of the optimization problem, imitating species evolutionary mechanism (Goldberg, 1989).

In an optimization problem, there is a function to optimize (cost function) and a zone where to look for (search space). Every point of the search space had an associated value of the function. The different points of the search space are the different individuals of population. Similarly to natural genetic, every different individual is characterized by a chromosome and in the optimization problem, this chromosome is made by the point coordinates in the search space.

The cost function value for an individual has to be understood as the adaptation level to the environment for such individual.

Evolutionary mechanism, that is, the rules for changing populations throughout generations is performed by Genetic Operators. A general Genetic Algorithm evolution mechanism could be described as follows:

From an initial population (randomly generated), the next generation is obtained as:

1. *Some individuals are selected for the next generation. This selection is made depending on adaptation level (cost function value). Such individuals with better J(x) value have more possibilities to be selected.*

2. *To explore search space, an exchange of information between individuals is performed by crossover. That produces a gene exchange between chromosomes. The rate of individuals to crossover is fixed by P_c, crossover probability.*

3.   *An additional search space exploration is performed by mutation. Some individuals are subject to a random variation in their genes. The rate of individuals to be mutated is set by mutation probability P_m.*

In this general framework, there are several variation in the Genetic Algorithm implementation; different gene codification, different genetic operator implementation, etc. Implementation for the present work has the following characteristics:

1.   Real value codification, each gene has a real value, the interpretation of the chromosome has been detailed in previous section.
2. $J(\mathbf{x})$ is not directly used as cost function. A linear 'ranking' operation is performed (Bäck, 1996). Ranking operation prevents the algorithm from exhausting, it avoids clearly dominant individuals prevailing too soon.
3.   Selection is made by the operator known as *Stochastic Universal Sampling (SUS)* (Baker, 1987).
4.   For crossover it is used *intermediate recombination* operator (Mühlenbein et al., 1993). Chromosomes sons (x'$_1$ and x'$_2$) are obtained through following operation on chromosomes fathers (x$_1$ and x$_2$):

$$x'_1 = \alpha_1 \cdot x_1 + (1-\alpha_1)\, x_2 \;;\; x'_2 = \alpha_2 \cdot x_2 + (1-\alpha_2)\, x_1 \;;$$
$$\alpha_1, \alpha_2, \in [\text{-}d,\ 1{+}d]$$

α_1 and α_2 have to be generated for each gene increasing search capabilities but with a higher computational cost. Implemented Genetic Algorithm has been adjusted as follows: $\alpha_1 = \alpha_2$ and generated for each chromosome, $d = 0$ and $Pc = 0,8$.
5.   Mutation operation is done with a probability $Pm = 0,1$ and a normal distribution with standard deviation set to 20% of search space range.

The high computational cost of Sonic Crystal optimization problem produces huge execution time, i.e. in a standard execution (population of 360 individuals, 250 generations) time is around 104 hours. Improvements of execution time have been obtained with a parallel implementation of the Genetic Algorithm described. Several alternative for parallelization are possible

(Cantú-Paz, 1995) the selected one is the configuration Master-Slave. For this architecture there is one processor working as Master, executing tasks of the Genetic Algorithm (ranking, selection, crossover and mutation), and the rest evaluate fitness function of a subpopulation (see Fig. 2).

The Master has to send subpopulation to each Slave, who makes fitness evaluation and returns results to the Master. The Master works in a synchronous way, waiting for all fitness value from all Slaves. After receiving all fitness values the Master performs the evolution to produce the next generation (genetic operators are executed) and sends to the Slaves the new population for fitness evaluation. This type of implementation is the most simple and does not change Genetic Algorithm operators and behaviour. The time reduction is significative since the overall time is divided by the number of Slaves. For the problem proposed, with 5 Slaves, the total execution is reduced to 21 hours.

All developments (Genetic Algorithm and Sonic Crystals models) have been made in Matlab®, parallelization has been done using Matlab Distributed Computing Toolbox and Matlab Distributed Computing Engine.

RESULTS

In this point we present some of our main results. In this work we have analyzed width ranges of 600 Hz

Figure 2. Master/slave architecture for parallel genetic algorithm
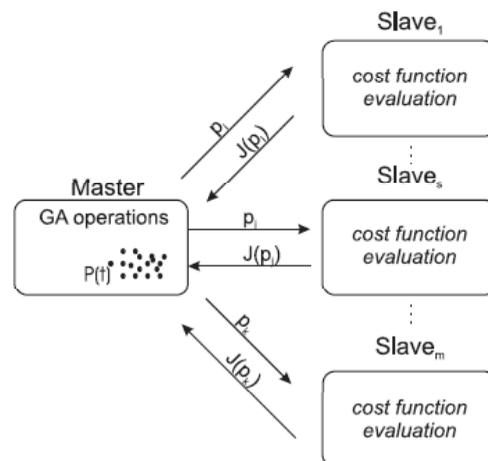
centered at several frequencies (800, 1100, 1300, 1700, 2000, 2300, 3090 Hz) above the first Bragg's peak. On the Fig. 3 we present the results corresponding to the ranges of frequencies centered at 1700 and 3090 Hz respectively. On the left hand of the Fig. 3 we present the schemes of cylinders of the Quasi Ordered Structures generated by the design tool described above. On the right hand the acoustic attenuation spectra calculated by the MST for the starting Sonic Crystals (continuous line) and for the optimized Quasi Ordered Structures (dashed line) is shown.

The creation of attenuation peaks in ranges of frequencies independents on the geometry of the starting Sonic Crystals using rigid scatterers has been the goal of this paper. As one can see on the Fig. 3, the peak attenuation in the spectra of the optimized Quasi Ordered Structures appears in the chosen frequency range, and this peak is absent in the spectra of the starting Sonic Crystals. Notice that the acoustic attenuation level in the frequency range in the starting Sonic Crystals is much lower than the Quasi Ordered Structures one. Even in some case the starting Sonic Crystals produces sound reinforcement. Moreover, the total number of cylinders in the optimized Quasi Ordered Structures is also lower than the starting Sonic Crystals one. In our results the number of cylinders is ranged between 36.7 % and 60%.

Figure 3. Optimized Quasi Ordered Structures and its spectrum. On the left hand the plot presents the schemes of cylinders of the optimized Quasi Ordered Structures. On the right hand the plots show the acoustic attenuation spectra calculated by the MST for the starting Sonic Crystals (continuous line) and for the optimized Quasi Ordered Structures (dashed line). (a) Optimization corresponding to the central frequency of 1700 Hz. (b) Optimization corresponding to the central frequency of 3000 Hz.

(a)

(b)
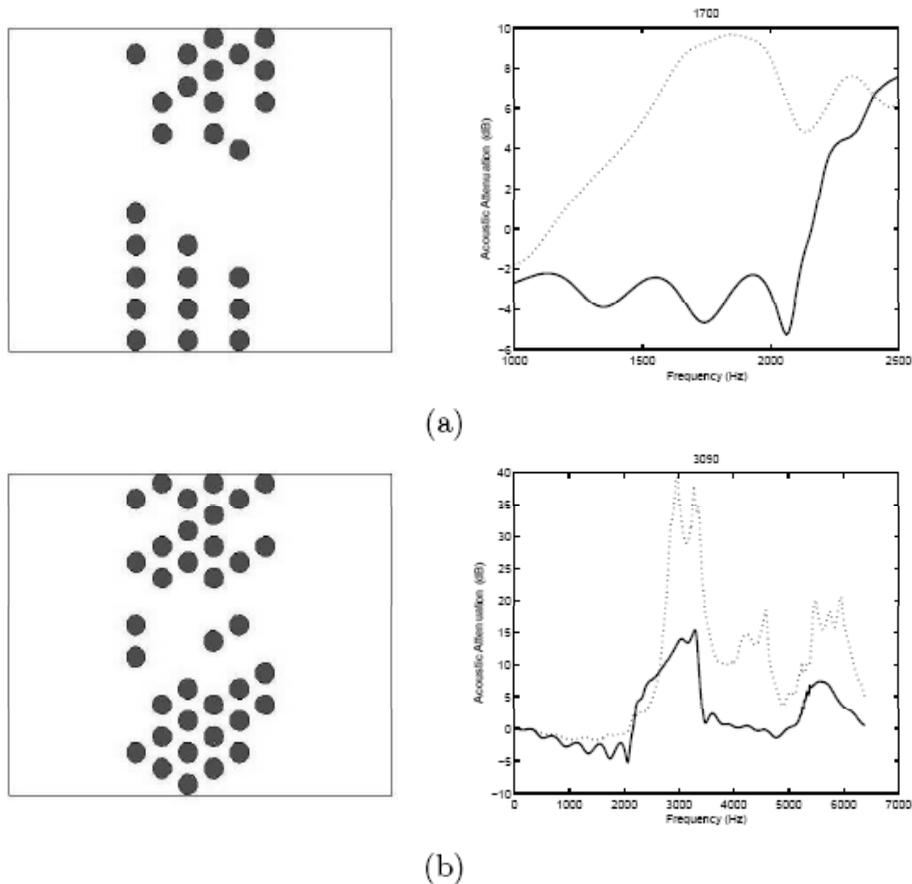
These results constitute a useful tool to design acoustic barriers based on Sonic Crystal with no need for sophisticated scatterers. The technological advantages of using Quasi Ordered Structures with rigid cylinders as scatterers are: high resistance for use outdoors, constructive simplicity and low cost due to the reduction in volume of the crystal.

CONCLUSION

This work shows an important and successful application of a Genetic Algorithm with a parallel implementation. Sonic Crystals open the way for innovative application in noise reduction in several interesting areas as acoustic noise barriers for traffic or general devices for controlling the noise. The Genetic Algorithm demonstrates an adequate optimization for a so complex problem and with the parallel implementation execution times are drastically reduced. Moreover, this method offers the possibility to test a wide range of Sonic Crystals adjustment in a reasonable time.

ACKNOWLEDGMENT

REFERENCES

T. Bäck. *Evolutionaty Algorithms in theory and practice*. Oxford University Press, New York, (1996).

J.E. Baker. Reducing bias and inefficiency in the selection algorithm. In Proc. Second International Conference on Genetic Algorithms, (1987).

D. Caballero, J. Sánchez-Dehesa, R. Martínez-Sala, C. Rubio, J.V. Sánchez Pérez, L. Sanchis and F. Meseguer. *Suzuki phase in two-dimensional sonic crystals*. Phys. Rev. B (64), 064303. (2001)

E. Cantú-Paz. *A summary of resaearch on parallel genetic algorithms*. Technical Report 95007, Illinois Genetic Algorithms Laboratory. IlliGAL, (1995).

Y.Y. Chen and Zhen Ye. *Theoretical analysis of acoustic stop bands in twodimensional periodic scattering arrays*. Phys. Rev. E (64), 036616(2001)

E.N. Economou and M.M. Sigalas. *Classical wave propagation in periodic structures: Cermet versus network topology*. Phys. Rev. B, (48), 18 ,(13434), (1993).

D.E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, (1989).

A. Hakansson, J. Sánchez-Dehesa and L. Sanchis. *Acoustic lens design by genetic algorithms*. Phys. Rev. B (70), 214302 (2004).

X. hu, C.T. Chan, and J. Zi. *Two dimensional sonic crystals with Helmholtz resonators*. Phys. Rev. E (71),055601 (2005).

M.S. Kushwaha, P. Halevi, G. Martínez, L. Dobrynski and B. Djafari-Rouhani. *Theory of acoustic band structure of periodic elastic composites*. Phys. Rev. B, (49),4, pp.2313-2322, (1994).

Z.Liu, X. Zhang, Y. Mao, Y.Y. Zhu, Z. Yang, C.T. Xhan, and P. Sheng. *Locally resonant sonic materials*. Science,(289), 1734, 2000.

R. Martínez-Sala, J. Sancho, J. V. Sánchez Pérez, J. Llinares, F. Meseguer. *Sound atteuntaion by sculpture*. Nature (London) (387), 241 (1995).

H. Mühlenbein and D. Schlierkamp-Voosen. *Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization*. Evolutionary Computation, (1), 1, (1993).

V. Romero-García, E. Fuster, L.M. García-Raffi, E.A. Sánchez-Pérez, M. Sopena, J. Llinares, J.V. Sánchez-Pérez. *Band gap creation using quasiordered strutures based on sonic crystals*. Appl. Phys. Lett., (88), 174104-1 174104-3, 2006.

M.M. Sigalas, E.N. Economou and M. Kafesaki. *Spectral gaps for electromagnietic and scalar waves: Possible explanation for certain differences*. Phys. Rev. B, (50), 5, (1994), (3393).

O

O. Umnova, K. Attenborough, and C.M. Linton. *Effects of porous covering on sound attenuation by periodic arrays of cylinders*. J. Acoust. Soc. Am. (119), 278 (2006)

KEY TERMS

Acoustic Attenuation Spectrum: Representation of the attenuation contribution of each acoustic frequency to a sound.

Cost Function: Mathematical function to minimize in an optimization problem.

Evolutionary Mechanism: Mechanism guided by biological evolution which represents the rules for changing populations throughout generations.

Filling Factor: Volume fraction occupied by the scattering material. Defined as, $ff=V_s/V$, where V is the total volume of the composite, and V_s the volume of the scattering material.

Genetic Algorithm: Global search method based on a simile of the natural evolution.

Quasi Ordered Structure: Given a starting Sonic Crystal (see Sonic Crystal), a quasi ordered structure (Quasi Ordered Structures) is the configuration of scatterers resulting of the creation of vacancies in the Sonic Crystal.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in.

Sonic Crystal: Arrays of scatterers placed periodically in space whose physical properties are different to the surrounding material.

Particle Swarm Optimization and Image Analysis

Stefano Cagnoni
Università degli Studi di Parma, Italy

Monica Mordonini
Università degli Studi di Parma, Italy

INTRODUCTION

Particle Swarm Optimization (PSO) is a simple but powerful optimization algorithm, introduced by Kennedy and Eberhart (Kennedy 1995). Its search for function optima is inspired by the behavior of flocks of birds looking for food.

Similarly to birds, a set (swarm) of agents (particles) fly over the search space, which is coincident with the function domain, looking for the points where the function value is maximum (or minimum). In doing so, each particle's motion obeys two very simple difference equations which describe the particle's position and velocity update.

A particle's motion has a strong random component (exploration) and is mostly independent from the others'; in fact, the only piece of information which is shared among all members of the swarm, or of a large neighborhood of each particle, is the point where the best value for the function has been found so far. Therefore, the search behavior of the swarm can be defined as emergent, since no particle is specifically programmed to achieve the final collective behavior or to play a specific role within the swarm, but just to perform a much simpler local task.

This chapter introduces the basics of the algorithm and describes the main features which make it particularly efficient in solving a large number of problems, with particular regard to image analysis and to the modifications that must be applied to the basic algorithm, in order to exploit its most attractive features in a domain which is different from function optimization.

BACKGROUND

One of the most attractive features of PSO, apart from its effectiveness and robustness with respect to local minima, is certainly its simplicity, which makes it trivial to implement in any programming language. It is also very versatile and applicable to a large number of optimization problems, virtually to any problem defined within a space for which a metric can be defined. However, its behavior, which mainly depends on the values of three constants, is still far from being fully understood. Extensive work (Engelbrecht2005, Clerc2006, Poli2007a) has provided very important insights into the properties of the algorithm, in studies where the dynamic properties of the swarm have been studied, even if under some restrictive assumptions.

The model which underlies PSO describes the motion of a swarm of particles within the domain of a function, usually termed *fitness function* as for evolutionary algorithms (Eiben 2004, de Jong 2006), seeking for its optimum. Such a motion is comparable to the random motion of a set of independent non-interacting particles within a force field generated by two attractors, one of which is specific to each cell.

The basic PSO equations for a generic particle P within the swarm are

$$\underline{X}_P(t) = \underline{X}_P(t\text{-}1) + \underline{v}_P(t) \tag{1}$$

$$\underline{v}_P(t) = \omega * \underline{v}_P(t\text{-}1) + C_1 * \text{rand}() * [\underline{X}_{Pbest} - \underline{X}(t\text{-}1)] + C_2 * \text{rand}() * [\underline{X}_{gbest} - \underline{X}(t\text{-}1)] \tag{2}$$

where $\underline{v}_P$ is the velocity of particle P, C_1 and C_2 are two positive constants, ω is the so-called inertia weight, $\underline{X}_P$ is the position of particle P, $\underline{X}_{Pbest}$ is the best-fitness point reached by P up to time t-1, $\underline{X}_{gbest}$ is the best-fitness point found by the whole swarm, rand() is a random value taken from a uniform distribution in the interval [0,1].

In its motion, the swarm explores the space effectively, usually converging rapidly to the optimum,

even if its behavior is strongly dependent on the values of ω, C_1, and C_2, which must be therefore set very accurately.

PARTICLE SWARM OPTIMIZATION AND IMAGE ANALYSIS

Even if much is still to be learned and discovered about PSO from a theoretical point of view (Kennedy 2007), as regards applications PSO is gaining more and more popularity. As reported in (Poli2007b), a very recent in-depth review of the field, searching the IEEExplore (http://ieeexplore.ieee.org) technical publication database by the keyword PSO returns a list of much more than 1,000 titles, about one third of which deal with theoretical aspects. This means that, to date, an incomplete list of PSO application papers adds up to little less than 1,000. Amazingly, about two thirds of them have been published in the last two years.

Image analysis is one of the fields to which PSO is being applied most frequently. As shown by a large number of papers in the image processing and computer vision literature, image analysis problems can be often reformulated as optimization problems, in which an objective function, directly derived from the physical features of the problem, is either maximized or minimized. In most cases, an optimum set of parameters which define the solution are sought using an optimization method. For most real-world problems, usually severely affected by noise or by the natural variability of the instances of the objects which must be detected, this is often inevitable, since methods in which closed-form solutions are directly applied are not usually robust enough with respect to such features. A large number of examples of applications of both traditional and evolutionary optimization methods including, as such, PSO, are reported in the literature.

In this section we will not consider direct applications of PSO as optimizers for an objective function. We will focus our attention on applications in which PSO is not only a way to 'tune' a more general algorithm by adapting it to the specific features of the problem at hand, but is directly part of the solution.

We will first introduce some general considerations on image analysis problems, which define the requirements imposed by them. This will allow us to reformulate some typical classes of problems encountered in image analysis, such as object detection and tracking or image segmentation, to include PSO, or some adapted version of its basic formulation, into the solution. We will then briefly show two examples of applications of PSO to segmentation and object detection, in which the above mentioned considerations have been taken into account.

PSO for Object Detection and Segmentation

In considering the application of PSO to image analysis tasks, one could assume the swarm to fly over the image to detect points or regions of interest. Therefore, the domain of the fitness function becomes the image itself. The fitness value to be assigned to each point can then be defined as a local function of image intensity in a neighborhood of that point, returning high values in points where features similar to the ones which are sought are found.

However, more global information must usually be extracted in image analysis tasks. In fact, while the basic PSO algorithm aims at finding a single optimum within the fitness landscape under exploration, in several image analysis applications more than one optimum (multiple objects) are to be found. This situation is typical of object recognition tasks, where the goal is to identify all possible occurrences of an object of interest characterized by a set of specific features. Similarly, in region-based segmentation, several regions with homogeneous features must be accurately located. Such requirements, encountered also in many other application areas, have led to the definition of several variants of PSO, in which particles are subdivided into a predefined number of sub-swarms, based on some clustering technique (Kennedy 2000, Veenhuis 2006, Passaro 2008), or through speciation (Chow 2004, Bird 2006, Leong 2006, Yen 2006), to achieve a dynamical reconfiguration of the swarm and the detection of an arbitrary number of regions of interest within the search space.

The velocity update function must also be modified in order to let the swarm spread as uniformly as possible over a whole area of interest featuring high fitness values. Such modifications may include introducing repulsive forces between particles, to prevent the whole swarm from converging onto the same point, and limiting particles' mobility inside a region of interest, to keep the swarm compact and in a stable configuration.

We will first show how these ideas can be applied to two common image analysis problems: region segmentation and object detection. Then we will show results obtained in two real-world problems: the first one was proposed as topic for a competition at GECCO 2006, and consists of detecting and segmenting as precisely as possible large pieces of pasta imaged over a set of noisy backgrounds over which also tiny pasta pieces are scattered, which must be ignored (see Figure 1). The second problem is a sub-task of plate recognition, in which the region occupied by a license plate is to be located within an image (see Figure 2). Even if the two tasks are semantically different, they share some common lower-level features, which allow the same modifications to basic PSO to be used in both cases, with a two-step approach. In the basic step, the image is explored, to focus on regions where interesting features are detected, before a refinement occurs in the subsequent step.

Modified PSO Equations for Image Analysis

In basic PSO, the fitness function is evaluated point by point. In analyzing images using PSO, the search space being the image, using such a local fitness function would make the search extremely sensitive to noise and possibly misleading. If fitness evaluation were just pixel-based, a meaningless isolated pixel yielding high fitness as a result of noise could attract and trap the whole swarm into its neighborhood.

To allow PSO to produce a uniform distribution of particles over each region of interest, the basic PSO algorithm can be modified in two directions:

- Forcing division of the swarm into sub-swarms, able to converge towards different regions of interest,
- Favoring dispersion of the particles all over the regions of interest.

Using the so-called *K-means PSO* (Passaro 2008), in which clusters of particles form based on their proximity within the search space, the former goal can be achieved. To achieve the latter, both the fitness function and the velocity-update equation must be modified.

As concerns the fitness function, a *local fitness* term, which evaluates how "interesting" the neighborhood of one pixel is, can be added to a *punctual fitness* function

term, whose value is computed based only on information carried by the pixel under consideration:

$fitness(x,y) = punctual\_fitness(x,y)$
$+ local\_fitness(x,y)$

The *local_fitness* term depends on the number of particles, with high punctual fitness, which are neighbors of the pixel located in *(x,y)*, and is given by:

$local\_fitness = K_0 * number\_of\_neighbors$

where *number_of_neighbors* is the number of particles within a pre-defined neighborhood of *(x,y)* and K_0 is a constant.

This way, the particles are attracted towards the areas where more pixels meet the punctual requirement, keeping away from isolated noisy pixels. This modification enhances the density of particles in the most interesting regions. To cover the whole extension of these regions, also the basic PSO velocity-update equation needs to be modified from (1) to:

$\underline{v}_P{}^*(t) = \underline{v}_P(t) + \underline{repulsion_P}$

The repulsion term can be expressed as

$|\underline{repulsion}(i,j)| = REPULSION\_RANGE - |\underline{X}_i - \underline{X}_j|$

where *i* and *j* are the particle indices and REPULSION_RANGE is the maximum distance within which the particles interact. Values of *repulsion(i,j)* are set to 0 for distances between *i* and *j* larger than REPULSION_RANGE. The global repulsion term *repulsion_P* for particle P is the average of all repulsion terms acting on it

$\underline{repulsionP} = (\ \sum_{j=1,N} \underline{repulsion}(P,j)\ )\ /\ n$

N being the number of particles in the swarm and *n* the number of particles within the neighborhood of P defined by REPULSION_RANGE.

Finally, to produce more stable sub-swarms, a particle with high punctual and local fitness is allowed to stand still with a probability which is linearly dependent on the particle density in its neighborhood, estimated as

$P\{v_P(t) = 0\} = n/N$

REAL-WORLD EXAMPLES

Pasta Segmentation

In a color-based region segmentation problem, the fitness function measures the similarity of the pixel color to the expected color of the objects of interest. For pasta, it can be expressed as:

if $(|r(x,y)-g(x,y)| < 30$ and $r(x,y)-b(x,y) > 60)$
then *punctual_fitness* $= 30 - |r(x,y) - g(x,y)|$
else *punctual_fitness* $= 0$

where $r(x,y)$, $g(x,y)$ and $b(x,y)$ are the red, green, and blue values, respectively, of the pixel located in *(x,y)*.

Since the goal is to obtain an accurate segmentation, up to pixel precision, and given the large number of pixels belonging to the objects of interest, PSO cannot obviously produce the final solution directly. Instead, it can be used in a pre-processing stage preceding a final thresholding stage which produces the actual output.

Following the PSO rules modified as previously described, the particles will tend to move towards larger pasta regions and stay around there. If one performs a number of PSO runs, assigning to each pixel a score which is directly proportional to the number of times a particle walks through it, the probability of belonging to a large pasta piece can be estimated for each pixel. To better estimate such a probability, avoiding bias deriving from the initial particle locations, each run should start with a different random initialization of the whole swarm. Image regions which eventually have high density of high-score pixels correspond to pieces of pasta. The final result of this stage, that we termed *global search*, is a preliminary segmentation by which the areas where large pieces of pasta are most likely to be found are grossly detected. To refine the segmentation, an algorithm which is very similar to the one used in the previous stage is applied; this time the domain where the swarm can move is limited to smaller regions surrounding pixel clusters whose score was above a threshold in the last phase of the global search. The final segmentation is eventually obtained

Figure 1. Pasta segmentation. Top: Original image (left) and results of global search (right). Bottom: Results of local search (left) and final segmentation (right).
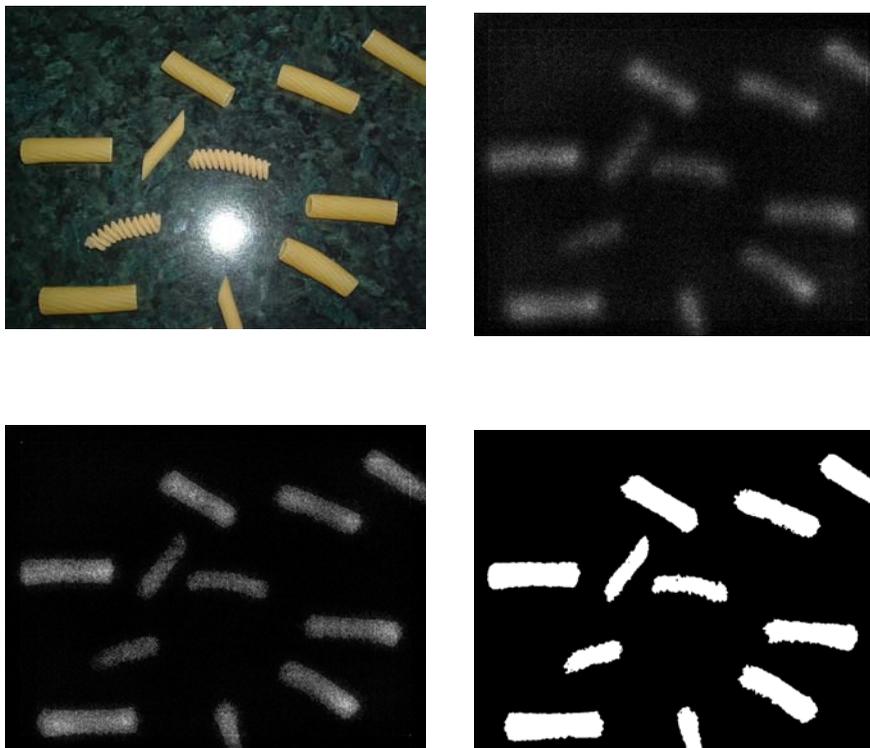
by thresholding the locally updated scores to obtain a binary image. Figure 1 shows the results obtained on one of the images from the image set used in the competition.

Plate Detection

In the license plate detection problem, the low-level feature on which detection is based is the density of high-level values of the horizontal gradient, due to the presence, in the plate, of symbols or symbol elements, which can be encountered when the image is scanned row-wise. Since a color image is available, we can use both color and gradient information, by first considering only those pixels which satisfy the typical features of plates (black characters on a white background for the most recent European standards), and then considering gradient information.

The punctual fitness of a pixel is defined as:

if (|r(x,y) - g(x,y)| > 30 or |r(x,y) - b(x,y)| > 30 or |g(x,y) - b(x,y)| > 30)
     punctual_fitness = 0;
else
     {right_gradient = |intensity(x,y) - intensity(x+1,y)|;
           left_gradient = |intensity(x,y) - intensity(x-1,y) |;
       if (right_gradient > left_gradient)
     punctual_fitness = right_gradient;
       else punctual_fitness = left_gradient; }

The basic PSO step is virtually the same as in pasta segmentation. However, a different algorithm is used, divided, as well, into a global and a local exploration stage in which, after the most promising areas are firstly located, the exploration is then refined to determine whether they actually include a plate.

In the global search, the swarm flies over the image until at least one sub-swarm of size greater than a pre-fixed threshold (50% of the whole swarm) has formed or a given number of iterations has been reached.

Then a local search is performed within regions where sub-swarms of sufficient dimension have formed, starting from the region occupied by the largest swarm; during this second stage: (i) the search is restricted to smaller image regions of interest enclosing the sub-swarms, (ii) the search is re-initialized activating a new full-size swarm in the region of interest, and (iii) the search is run for a pre-set number of iterations. At the end of this stage, a new bounding box, containing all particles, is computed. If this box has an aspect ratio compatible with a license plate, the plate is considered to have been found. Otherwise, the swarm is expanded along its two dimensions, by forcing low-fitness particles to move only horizontally or vertically, in order to reach higher-fitness points and, possibly, to let the bounding box reach the expected aspect ratio; in case of failure, the current region is discarded and the next area detected during the global search is explored.

Figure 2 shows the original image, along with the results of the global and local search, and the final result of the PSO-based algorithm.

The algorithm is computationally very efficient. A number of function evaluations is required to detect the plate, which is lower than just computing the whole gradient image, which would be just the very first step in any 'traditional' computer vision approach. Iteratively re-initializing, in each frame, the swarm location in a neighborhood of the region where the plate has been detected in the previous one, real-time performances can be achieved in tracking the plate in videos acquired at 30 frames per second using a standard PC.

Figure 2. License-plate detection; Original image (left) and results of the detection (right)

The same cannot be said for the pasta segmentation algorithm if high segmentation accuracy is required (about 30 seconds were needed to produce the segmentation in Figure 1 on a 2.8 GHz PC). However, even in that case, if the pieces are just to be grossly located, just a few runs of the algorithm are enough to achieve the goal.

FUTURE TRENDS

Research on PSO and PSO applications to the most various fields is booming nowadays. Image analysis is no exception: according to the INSPEC bibliography database, the number of papers which describe applications of PSO to such a field has increased by almost 50% in the last six months. Results are already very encouraging and suggest that much more is to be expected in the near future.

CONCLUSION

PSO is a versatile and effective optimization technique whose features can be easily adapted to a vast variety of problems, in solving which it can act not only as a "plain" optimizer, but as a more general, flexible search paradigm. The applications described in this chapter have confirmed this, introducing a general framework which can be applied, with few changes, to many other object detection and recognition problems, as well as to other lower-level tasks in computer vision, such as image segmentation.

REFERENCES

Bird, S. & Li, X. (2006). Enhancing the robustness of a speciation-based PSO. Proc. IEEE Congress on Evolutionary Computation, 3185-3192.

Chow, C.K. & Tsui, H.T. (2004). Autonomous agent response learning by a multispecies particle swarm optimization. Proc. IEEE Congress on Evolutionary Computation, 778-785.

Clerc, M. (2006). Particle Swarm Optimization. ISTE.

De Jong, K.A. (2006). Evolutionary Computation: a unified approach. MIT Press.

Engelbrecht, A.P. (2005). Fundamentals of Computational Swarm Intelligence. Wiley.

Eiben, A. & Smith, J. (2004). Introduction to Evolutionary Computation. Springer.

Kennedy, J. & Eberhart, R. (1995). Particle Swarm Optimization. Proc. IEEE International Conference on Neural Networks. 1942-1948, Vol. IV.

Kennedy, J. (2000). Stereotyping: improving particle swarm performance with cluster analysis. Proc. IEEE Int. Conference on Evolutionary Computation, 1507-1512.

Kennedy, J., Poli, R. & Blackwell, T. (2007) Particle Swarm Optimisation: an overview. Swarm Intelligence, in press.

Leong, W.F. & Yen, G.G. (2006) Dynamic population size in PSO-based multiobjective optimization. Proc. IEEE Congress on Evolutionary Computation, 6182-6189.

Passaro, A. & Starita, A. (2008) Particle swarm optimization for multimodal functions: A clustering approach. *Journal of Artificial Evolution and Applications, Volume 2008, Article ID 482032*.

Poli, R. (2007) The sampling distribution of particle swarm optimizers and their stability. Tech. Rep. CSM-465, Department of Computer Science, University of Essex.

Poli, R. (2007). Analysis of the publications on the applications of Particle Swarm Optimisation. Journal of Artificial Evolution and Applications, in press.

Veenhuis, C. & Köppen, M. (2006) Data swarm clustering. In Abraham, A., Groşan, C. & Ramos, V. (eds.). Swarm Intelligence in Data Mining. Springer, 221-241.

Yen, G.G. & Daneshyari, M. (2006). Diversity-based information exchange among multiple swarms in Particle Swarm Optimization. Proc. IEEE Congress on Evolutionary Computation, 6150-6157.

P

KEY TERMS

Evolutionary Computation: Collection of techniques, basically aimed at function optimization but applicable to a huge variety of problems, by which the optimum of a function (fitness function) is sought through iterative refinements, according to rules inspired by the laws of natural evolution.

Fitness Function: In evolutionary computation, the objective function which is to be optimized.

Image Analysis: Collection of techniques by which high-level information content is extracted from a digital image using image processing and computer vision techniques.

Particle Swarm Optimization: Optimization technique inspired by the exploratory behavior of animal swarms/flocks/herds in search of food.

Segmentation: In computer vision, a process by which an image is subdivided into regions having homogeneous visual features.

Sub-Swarm: In particle swarm optimization, subset of a swarm, within which the distance between any particle and the closest one is below a pre-set threshold.

Swarm Intelligence: Collection of techniques, usually inspired by nature, in which high-level intelligent behaviors emerge as a result of the interaction among a high number of agents which, individually, perform apparently trivial, low-level tasks.

Personalized Decision Support Systems

Neal Shambaugh
West Virginia University, USA

INTRODUCTION

Decision support systems (DSS) are computerized systems that assist humans to make decisions. Early versions were designed for executives, but over time DSSs were designed for workers at any level in the organization (Keen & Morton, 1978; Rockart, 1979). Due to increasing costs in providing benefits and services, organizations are forcing workers and consumers to take increasing responsibility for insurance, health care, and financial planning decisions. Extreme events, such as terrorism, pandemics, and natural disasters will swamp the capacity of governmental agencies to serve their citizenry. Individuals in affected communities must turn to local agencies or ad hoc groups for assistance. **Personal decision support systems** (PDSS), consisting of databases, model-based expertise, and intelligent interfaces, along with wireless communications, Internet resources, and personal computing, provide sufficient resources to assist informed individuals and groups in solving problems.

This article reviews the typical components of a DSS and the different types of systems that have evolved. The article poses three types of problems facing individuals, including routine problem solving, immediate survival needs, and long-term evolutionary growth. Personal decision support issues of acquiring information, processing information, and dissemination are outlined. Future trends and research opportunities are discussed.

BACKGROUND

DSS aid human thinking by accessing information, integrating this information in some way, structuring decisions, and optimizing decisions (Sprague & Carlson, 1982). These benefits are obtained using three major system features of a DSS, which include a database, which records knowledge; a model base, which models or represents expertise and problem-solving; and an interface, which provides a user with a means to interact with the other system components (Sprague, 1980).

Powers (2007) characterized DSS in terms of how the system provides assistance. Model-driven DSSs for individuals include spreadsheets. Data-driven DSS, such as **Executive Information Systems** (EIS), are used by organizations and institutions for strategic and tactical decisions. Communication-driven DSSs can be seen in groupware, video conferencing, and bulletin boards. A document-driven DSS, such as provided by search engines, facilitates document retrieval. A knowledge-driven DSS would be used to solve specialized problems and consist of knowledge represented in terms of rules, procedures, hierarchical frames, or networks. Most recently, web-based DSSs are found in browser searching, intranets, and portal use.

Decision support systems are based on the notion that human reasoning is a rational process, although this is not always the case particularly when humans are faced with complexity and stress (Druzdzel & Flynn, 2000). Experts' decisions in real settings have been shown to demonstrate less quality than linear models (Hastie & Dawes, 2001). Judgmental heuristics reduce cognitive load but decrease the quality of decisions. Characteristics of the DSS components vary in a PDSS in order to compensate for the type of problems faced by individuals. In general for a PDSS the data bases are customized, the model bases are organized along preferential outcomes (e.g., more or less, quantitative), decisions (e.g., lists and value ordering), and uncertainty (specific actions resulting in gain considering constraints and price).

PERSONALIZED DECISION SUPPORT

This article summarizes three problem types facing individuals, including routine problem solving, extreme survival needs, and long-term change. The article outlines system architecture requirements in terms of acquiring and processing of information, interacting

with this information, and the dissemination of information and recommendations.

PDSS Problem Types

The consumer of the 21st century faces numerous **routine problems**, such as career choice, self-improvement, volunteerism, financial planning, retirement, insurance, consumer purchases, health care physician, and personal health. PDSS applications can be seen in health care ranging from point-of-care use of personal data assistants (PDA) to helping patients make decisions on health care (Crawford, 1997; Pierce, 1998). Routine problems consist of complex options with short-term benefits and unknown long-term implications. However, individuals tend to discount the need to make decisions and/or the belief that institutions and governmental agencies will impose decisions on them.

A second problem type can be classified as **survival**. Three examples include natural disasters, terrorism, and pandemics. Natural disasters, such as hurricanes, tornadoes, floods, drought, volcanic eruptions, earthquakes, and meteorite impacts, can also include gradual changes brought about by global warming. Radical changes could involve results of nuclear winter, the shift of the moon's orbit, or pole shifting of the earth's magnetic field. PDSS applications involve disaster management and attempts to connect satellite mapping technology with government agencies (Hegde, Srivastava & Manikiam, 2004). Terrorism provides a more recent survival problem brought about by racial cleansing, violence between religious groups, undermining of governments through corruption and assassination, chemical warfare, and destruction of neighbourhoods and infrastructure. PDSS applications for this problem type has emerged for counter-terrorism applications (Alward, 2004). Pandemics have always occurred throughout human history but have taken on serious implications given technological developments in genetics. Survival problems cannot be predicted, fully characterized, and their impact overwhelms the capacity of a DSS. The value of a PDSS is its proactive potential by identifying national, state, and local resources, recommending action, and triggering the development of institutional support and awareness that did not exist before.

A third problem type is evolutionary or long-term change brought about by a realization that existing decision paths may lead to significant consequences.

Awareness of **change problems** signal a need for people to make long-term proactive decisions in light of multiple paths or scenarios (Schellnhuber, Crutzen, Clark, Clausssen, & Held, 2004). Proactive decision-making enables humans to become aware of and address serious consequences of prior decisions by individuals, groups, institutions, and governments, as well as the impact of technological innovations. However, change problems tend to be low priority, require significant resources, and they resist consensus due to their apparent intractability. Simulations and virtual environments may be needed to help citizens interact with potential paths (Stanney, 2002).

Personal Decision Support Architecture

Early views defined a personal DSS as one which focused on a discrete task or decision (Rockart & Bullen, 1986). Examples frequently involved group support, such as Morton's (1971) DSS which involved both marketing and production planning. Keen and Hackathorn (1986) identified three main parts of a personal DSS to include the interface between machine and user, relevant operators (i.e., action verbs, such as "help"), and a database. Development of a personal DSS requires attention to dialogue, refinement of the vocabulary-operators, and evolution of the data structure of the database.

PDSS, as described here, would involve both individual and social needs, and thus would be hybrid versions of several DSS types (Powers, 2007). A PDSS would include mathematical and statistical tools (model-driven) to calculate and make inferences on numerical data. They would retrieve forms and information (document-driven) to support decision-making. They would use information and data as input to address specialized needs (knowledge-driven), such as health care, insurance, career options, and travel planning, among others. The PDSS would consist of both localized (personal computer system) resources and distributed (web-driven) sources where information and computing may be conducted at other sites.

The major systems of a PDSS include databases, reasoning models, interface, and communication options. Each of these four systems can be equated to acquiring information, processing this information in ways that make it amenable to specialized decision modules (e.g., insurance, health-care, travel planning), interacting with the information visually, and com-

Figure 1. PDSS system features

Acquiring	Processing	Interacting	Disseminating
Databases	Model bases	Dialogue-Interface	Communication
• Local, personalized • Remote, browsing • Remote, integrated	• Expertise • Specialized functions • Heuristic patterns • Context objects	• Desktop metaphor • Personal metaphor • Task-specific • Just-in-time	• Wireless • Internet posting • Internet feedback • Portals • Collaboration

municating or sharing decisions or information with others (see Figure 1).

Acquiring Information

Databases provide a repository for information within any DSS. A personalized version of a DSS would combine local databases, which are developed individually for specific needs, with remote integrated databases. These databases would consist of inconsistent structures, while in the long term some standardization of database structure would be required to develop a personalized integrated database. In addition, ad hoc browsing tends to characterize individual information needs with little regard for organizing this information over the long term.

Processing Information

One of the powerful features of a DSS is its model base. Modeling allows knowledge to be applied across problems and facilitates analysis, explanations, and advocacy (Druzdzel & Flynn, 2000). A model base would include one or more models or representations of expertise ranging from highly specialized (e.g., resale home value) to more general (e.g., model of a learner). Model bases might become object-oriented and incorporated into a PDSS like a software plug-in as needed. Generic versions of a PDSS might include a range of common model components for financial, employment, travel, and health needs and provide simulations to help a user see the implications of decisions. Integrating model bases, as with databases, will require some standardization of model structure along some common categories. Personal patterns of

reasoning may also be archived to provide speed and options for new problems.

The most important and the most challenging to archive and characterize would be context information, an example of unstructured data. A top-down version of a system that would increase the structure of the context-data would be to categorize specific routine contexts, such as financial, health, college selection. Extreme survival categories could include natural disaster and other types of emergencies, crime and terrorism, and pandemics. A bottom-up version of a context representation system would be to identify patterns of information using semantic webs (Hädrich & Priebe (2005), and over time a context-map would be built to characterize particular categories of context.

Interacting with Information

Human dialogue with databases and model bases has used a visual interface, which has typically featured a desktop metaphor. To date users have relied on the metaphor presented to them. A customized interface could still use a desktop metaphor to organize individual problem needs. Other options could be available and custom-developed, which might still rely on an inventory of choices or through some metaphor of choice. Specialized interfaces could be used depending on the problem type (routine, survival, change) to facilitate decision-making. Survival needs require that a user not be presented with too many choices, but rather accurate options to meet an immediate need. These just-in-time visual views present just the information and advice as needed (Lieberman, 2002).

Disseminating Information

The dissemination function, involving the communication and sharing of information and decision options with others, represents a critical system component of a PDSS. While routine problems relate to an individual, problems of survival and change require collaboration. Multi-point sharing of information facilitates decision-making. As wireless becomes a standard feature in many technological devices, dissemination and communication increases for more people. Wireless may become an antiquated term as it becomes transparent and common. Information can be posted for everyone or particular audiences and can be edited or linked to other sources. Much of this information and collaboration may become routed through **personal portals** which structure the information for other users (Shambaugh, 2007).

FUTURE TRENDS

Future Design Metaphor

One feature of a DSS includes the retrieval of information so that decisions can be made based on this information and other sources. Decisions are then based on existing data or data from the past. Goals of profit and cost reduction rely on what-if scenarios and simulations based on assumptions. The focus of individuals, however, is rarely on the past but on the present and the near future. Although the future cannot be predicted, trends based on past and current data provide a picture of where we are in our business, career, or personal life. Making decisions on what we want our life to be for ourselves, our families, and our communities, and even "what business are we in?" necessitates a different view that of **future design**, which is not about predicting the future but rather working towards a future based on our intent to continually cycle through rethinking, designing, and improving.

Government and Community

Responsibility for daily life has always been the domain of the individual and the family. However, the historical reality is that daily life has been continuously constrained by institutions and governments, and by the unseen consequences of technological innovation.

Much of daily life requires navigating these constraints and impacts. However, these tensions can be ameliorated with a move towards taking advantage of personal insight and motivation, a belief in taking responsibility for our lives and our communities, and designing our technological tools for where we want to go, all features of a future design stance.

Research Opportunities

One avenue for research is to add more structure to unstructured data, including information from remote sources, locally-developed databases, and context information. How might these different sources of information be integrated and generalized for use by others? How might context be characterized in terms of re-usable objects?

Modeling expertise has been a long-standing challenge in AI. Modeling decisions for routine problems, those that can be characterized by rules or procedures, and use static domain models, have been the most successful. But a bigger question beyond *What do we know?* becomes *How does the model update itself*?

Decision-making in survival situations will require customized model bases developed specifically for categories of extreme survival. In these type of situations problems are unique and tools will need to be developed see how users' beliefs about uncertainty and preferences on different outcomes can be visualized (Howard & Matheson, 1984). Evolutionary decision-making, decisions that impact long-term change, will require that model bases evolve from new data. Continually re-defining expertise provides opportunities to analyze what people do on a daily basis (Gigerenzer, Todd, & ABC Research Group, 1999) and how daily, routine expertise becomes critical for individuals and groups of individuals.

Furthermore, inquiry could be conducted on how informed citizens create new societies, **epistemic cultures** that are themselves creating new bodies of knowledge (Cetina, 1999). These new societies could be a block of families, an online community of individuals, or physical neighbourhoods, cities, or countries, or geographic regions. The idea of a PDSS does not limit itself to an individual but to personalizing human life as tools to help individuals, neighborhoods, and cities grow (Longworth, 2006). The conundrum for researchers and designers is realizing that in designing systems that are less logical and more approximations

of the messiness of real life they may be helping humans come to understand what it means to be human (Johnson, 2005).

Another research avenue would study how users might determine the user interface, based on personal metaphors or specific needs, rather than reacting to a standardized metaphor. The study of mental models and how humans project meaning from their experience to a new experience might provide a new means to think and act beyond old rules (Fauconnier & Turner, 2002). Not all problems and situations require the same interface, particularly as the severity of the problem may require a design focused on immediacy and limited choice. Continued collaboration between AI researchers who study representation and reasoning, and those in Human-Computer Interaction (HCI), in which interaction is addressed, may lead to intelligent interfaces with flexible planning, incorporation of human constraint issues (e.g., time, patience, attention, motivation, cognitive demands), and relevance of context (Lieberman & Selker, 2000). Such intelligent interfaces may find themselves first in wireless devices, such as PDAs.

CONCLUSION

Specific skills and responsibilities for living in the 21st century have been pushed down to consumers by organizations and governmental agencies. Individuals now require more time to make important decisions related to their personal and professional lives. These personal decisions add to the growing complexity of human living and require time and resources. Technological developments in computing, networking, and communication provide humans with the capacity for making informed decisions. With the prospect of survival threats and long-term change, informed groups of citizens can initiate proactive priorities in their national, state, and local governments to address these potential problems. A PDSS with features that enable communication and collaboration creates a tool to help individuals take responsibility for decision-making rather than relying on government and institutions. Personalized decision support, characterized by access to Internet resources, integrated knowledge bases, and personal computing and wireless communication, can provide humans with information and recommendations to solve problems, address emergencies, and enhance life.

REFERENCES

Alward, R. (2004). *Personal decision support aids for special operations, Report of Syndicate One*. Retrieved on August 30, 2007 from http://handle.dtic.mil/100.2/ADA427997.

Cetina, K. K. (1999). *Epistemic cultures: How the sciences make knowledge*. Cambridge, MA: Harvard University Press.

Crawford, P. (1997). Computer-assisted decision support in health care. *Annual Meeting of the International Society of Technology Assess Health Care Meeting, 13*, 170.

Druzdzel, M. J., & Flynn, R. R. (2000). Decision support systems in A. Kent (Ed.). *Encyclopedia of library and information science*, *67*, Suppl. 30 (pp. 120-133). New York: Marcel Dekker.

Fauconnier, G., & Turner, M. (2002). *The way we think: Conceptual blending and the mind's hidden complexities*. New York: Basic Books.

Gigerenzer, G., Todd, P. M., & ABC Research Group. (1999). *Simple heuristics that make us smart*. New York: Oxford University Press.

Hädrich, T., & Priebe, T. (2005), A context-based approach for supporting knowledge work with semantic portals. *International Journal of Semantic Web and Information Systems, 1*(3), pp. 64-88.

Hastie, R., & Dawes, R. M. (2001). *Rational choice in an uncertain world: The psychology of judgement and decision making* (2nd Rev. Ed.). Thousand Oaks, CA: Sage.

Hegde, V. S., Srivastava, S. K., & Manikiam, B. (2004). *Space resources, operational services, and future plans*. India-United States Conference on Space Science, Applications and Commerce. Retrieved on August 30, 2007 from http://www.aiaa.org/indiaus2004/Disaster-management.pdf.

Howard, R. A., & Matheson, J. E. (1984). Influence diagrams. In R. Howard & J. Matheson (Eds.). *The principles and applications of decision analysis*, 719-762, Menlo Park, CA: Strategic Decisions Group.

Johnson, M. (2005). Swamped by the updates: Expert systems, semiclasm, and apeironic education. In S.

Franchi & G. Guzeldere (Eds.). *Mechanical bodies, computational binds: Artificial intelligence from automata to cyborgs* (pp. 365-388). Cambridge, MA: MIT Press.

Keen, P. G. W., & Scott Morton, M. S. (1978). *Decision support systems: An organizational perspective*. Reading, MA: Addison-Wesley.

Keen, P. G. W., & Hackathorn, R. D. (1986). Decision support systems and personal computing. In J. F. Rockart & C. V. Bullen (Eds.). *The rise of managerial computing: The best of the Center for Information Systems Research, Sloan School of Management, MIT*. Homewood, Ill: Dow Jones-Irwin.

Lieberman, H. (2002). Interfaces that give and take advice. In J. M. Carroll (Ed.). *Human-computer interaction in the new millennium* (pp. 475-486). Boston, MA: Addison-Wesley.

Lieberman, H., & Selker, T. (2000). Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal, 39*(3 & 4), 617-631.

Longworth, N. (2006). *Learning cities, learning regions, learning communities: Lifelong learning and local government*. New York: Routledge.

Morton, S. (1971). *Management decision systems: Computer-based support for decision making*. Cambridge, MA: Division of Research, Harvard Business School.

Pierce, P. F. (1998). *Choices: An interactive decision support program for breast cancer treatment*. Retrieved on August 30, 2007 from http://handle.dtic.mil/100.2/ADA369255.

Powers, D. J. (2007). *A brief history of decision support systems*. DSS. Resources.COM, retrieved on March 10, 2007 from http://DSSResources.COM/history/dsshistory.html.

Rockart, J. F. (1979). Chief executives define their own data needs. *Harvard Business Review, 67*(2), 81-93.

Rockart, J. F. & Bullen, C. V. (1986). *The rise of managerial computing: The best of the Center for Information Systems Research, Sloan School of Management, MIT*. Homewood, Ill: Dow Jones-Irwin.

Schellnhuber, H. J., Crutzen, P. J., Clark, W. C., Claussen, M., & Held, H. (2004). *Earth system analysis for sustainability*. Cambridge, MA: MIT Press.

Shambaugh, N. (2007). Personal portals. In A. Tatnall (Ed.). *Encyclopedia of portal technologies and applications*. Hershey, PA: IGI Global.

Sprague, R. H., Jr. (1980). A framework for the development of decision support systems. *Management Information Systems Quarterly, 4*(4), 1-26.

Stanney, K. M. (2002). *Handbook of virtual environments: Design, implementation, and applications*. Mahwah, NJ: Lawrence Erlbaum Associates.

KEY TERMS

Change Problems: A type of problem with long-term consequences.

Decision Support System (DSS): A computerized system which assists humans to make decisions.

Epistemic Cultures: Bodies of knowledge developed by individuals with a common need.

Executive Information System (EIS): A decision support system that directly supports management decisions.

Future Design: A means of looking and working towards the future rather than predicting the future.

Personal Decision Support System (PDSS): A computerized decision support system which acquires information and organizes the information so that models of reasoning can produce recommendations for further information, resources, or action. Another feature of PDSS is its capacity to openly communicate organized information or decisions to others.

Personal Portals: A computerized site which provides a gateway other sites of individual interest.

Routine Problems: A type of problem faced by individuals involving complexity of choices as well as short-term and long-term implications.

Survival Problems: A type of problem characterized by extreme impacts on individuals and communities.

Planning Agent for Geriatric Residences

Javier Bajo
Universidad de Salamanca, Spain

Dante I. Tapia
Universidad de Salamanca, Spain

Sara Rodríguez
Universidad de Salamanca, Spain

Juan M. Corchado
Universidad de Salamanca, Spain

INTRODUCTION

Agents and Multi-Agent Systems (MAS) have become increasingly relevant for developing distributed and dynamic intelligent environments. The ability of software agents to act somewhat autonomously links them with living animals and humans, so they seem appropriate for discussion under nature-inspired computing (Marrow, 2000). This paper presents AGALZ (Autonomous aGent for monitoring ALZheimer patients), and explains how this deliberative planning agent has been designed and implemented. A case study is then presented, with AGALZ working with complementary agents into a prototype environment-aware multi-agent system (ALZ-MAS: ALZheimer Multi-Agent System) (Bajo, Tapia, De Luis, Rodríguez & Corchado, 2007). The elderly health care problem is studied, and the possibilities of Radio Frequency Identification (RFID) (Sokymat, 2006) as a technology for constructing an intelligent environment and ascertaining patient location to generate plans and maximize safety are examined.

This paper focuses in the development of nature-inspired deliberative agents using a Case-Based Reasoning (CBR) (Aamodt & Plaza, 1994) architecture, as a way to implement sensitive and adaptive systems to improve assistance and health care support for elderly and people with disabilities, in particular with Alzheimer. Agents in this context must be able to respond to events, take the initiative according to their goals, communicate with other agents, interact with users, and make use of past experiences to find the best plans to achieve goals, so we propose the development of an autonomous deliberative agent that incorporates a Case-Based Planning (CBP) mechanism, derivative from Case-Based Reasoning (CBR) (Bajo, Corchado & Castillo, 2006), specially designed for planning construction. CBP-BDI facilitates learning and adaptation, and therefore a greater degree of autonomy than that found in pure BDI (Believe, Desire, Intention) architecture (Bratman, 1987). BDI agents can be implemented by using different tools, such as Jadex (Pokahr, Braubach & Lamersdorf, 2003), dealing with the concepts of beliefs, goals and plans, as java objects that can be created and handled within the agent at execution time.

BACKGROUND

During the last three decades the number of Europeans over 60 years old has risen by about 50%. Today they represent more than 25% of the population and it is estimated that in 20 years this percentage will rise to one third of the population, meaning 100 millions of citizens (Camarinha-Matos & Afsarmanesh, 2002). This situation is not exclusive to Europe, since studies in other parts of the world show similar tendencies (Camarinha-Matos & Afsarmanesh, 2002). The importance of developing new and more reliable ways to provide care and support to the elderly is underlined by this trend (Camarinha-Matos & Afsarmanesh, 2002), and the creation of secure, unobtrusive and adaptable environments for monitoring and optimizing health care will become vital. Some authors (Nealon & Moreno, 2003) consider that tomorrow's health care institutions will be equipped with intelligent systems capable of

interacting with humans. Multi-agent systems and architectures based on intelligent devices have recently been explored as supervision systems for medical care for the elderly or Alzheimer patients, aimed to support them in all aspects of daily life, predicting potential hazardous situations and delivering physical and cognitive support.

RFID technology is a wireless technology used to identify and receive information on the move. An RFID system contains basically four components: tags, readers, antennas and software (Sokymat, 2006). The configuration used in the system presented in this paper consists of 125KHZ transponders mounted on bracelets worn on the patient's wrist or ankle, several readers installed over protected zones, with up to 2 meters capture range, and a central computer where all the ID numbers sent by the readers is processed.

MAIN FOCUS OF THE CHAPTER

This article presents an autonomous planner agent for health care. The autonomous nature-inspired health care agent, named AGALZ, is presented. Then, a case study is presented, describing the main characteristics of ALZ-MAS architecture and its agents, including AGALZ, finalizing with initial results obtained after the implementation of a prototype into a real scenario.

Autonomous Nature-Inspired Health Care Agent

We have developed AGALZ, an autonomous deliberative Cased-Based Planner (CBP-BDI) agent that integrates with other agents into a multi-agent system, named ALZ-MAS, as a proposal to improve the efficiency of health care and supervision of patients in geriatric residences. AGALZ presents a deliberative architecture, based on the BDI (Belief, Desire, Intention) model (Bratman, 1987). In this model, the internal structure and capabilities of the agents are based on human mental aptitudes, using beliefs, desires, and intentions. Our method facilitates the incorporation of CBR systems (Aamodt & Plaza, 1994) as a deliberative mechanism within BDI agents, facilitating learning and adaptation and providing a greater degree of autonomy than pure BDI architecture. A deliberative CBP-BDI agent is specialized in generating plans and incorporates a Case-Based Planning (CBP) mechanism. The

purpose of a CBR agents is to solve new problems by adapting solutions that have been used to solve similar problems in the past (Aamodt & Plaza, 1994), and the CBP agents are a variation of the CBR agents, based on the plans generated from each case. A CBP planner is used for AGALZ to find plans to give daily nursing care in a geriatric residence (Tapia, Bajo, Corchado, Rodríguez & Manzano, 2007). It is very important maintaining a map with the location of the different elements that take part in the system at the moment of planning or replanning, so using RFID technology facilitates enormously the dynamic planning.

CBR is a type of human thinking based on reasoning about past experiences. To introduce a CBR motor into a BDI agent it is necessary to represent the cases used in a CBR system by means of beliefs, desires and intentions, and implement a CBR cycle. A case is a past experience composed of three elements: an initial state or problem description that is represented as a belief; a final state that is represented as a set of goals and a solution (belief); and the sequence of actions that makes it possible to evolve from an initial state to a final state. This sequence of actions is represented as intentions or plans. Figure 1 shows the internal structure of a CPB-BDI agent.

In a planner agent, the reasoning motor generates plans using past experiences and planning strategies, so the concept of Case-Based Planning is obtained (Corchado & Laza, 2003; Glez-Bedia & Corchado, 2002). CBP consists of four sequential stages: retrieve stage to recover the most similar past experiences to the current one; reuse stage to combine the retrieved solutions in order to obtain a new optimal solution; revise stage to evaluate the obtained solution; and retain stage to learn from the new experience.

The CBP cycle is implemented through goals and plans. When the goal corresponding to one of the stages is triggered, different plans (algorithms) can be executed concurrently to achieve the goal. Each plan can trigger new sub-goals and, consequently, cause the execution of new plans. Deliberative CBP-BDI agents, like AGALZ, are able to incorporate other reasoning mechanisms that can coexist with the CBP. AGALZ is an autonomous agent that can survive in dynamic environments. However, is possible to incorporate communication mechanisms that allow it to be easily integrated into a multi-agent system and work coordinately with other agents to solve problems in a distributed way.

Figure 1. CBP-BDI Agent internal structure
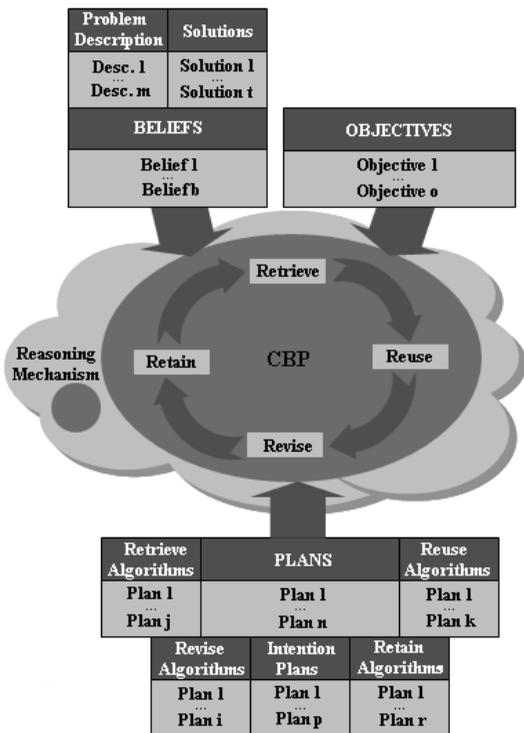
The CBP planner constructs plans in such a way that a plan is a sequence of tasks that need to be carried out by a nurse. A task is a java object that contains the date of the requested service, the description of the service and the time limits to carry it out.

For each task one or more goals are established, in such a way that that the whole task is eventually achieved. A problem description will be formed by the tasks that the nurse needs to execute, the resources available, and the times assigned for their shift. In the retrieve stage, those problem descriptions found within a range of similarity close to the original problem description are recovered from the beliefs base. In our case, a tolerance of 20% has been permitted. In order to do this, AGALZ allows the application of different similarity algorithms (cosine, clustering etc.). Once the most similar problem descriptions have been selected, the solutions associated with them are recovered. One solution contains all the plans (sequences of tasks) carried out in order to achieve the objectives of AGALZ for a problem description (assuming that replanning is possible) in the past, as well as the efficiency of the solution being supplied. The chosen solutions are combined

in the reuse stage to construct a plan (Bajo, Corchado & Castillo, 2006; Glez-Bedia & Corchado, 2002). The reuse is focused on the objectives and resources needed by each task, as well as on the objectives that the nurse needs to perform and the resources available in order to carry out the global plan. The objectives that each nurse has are aimed to attend the patients and not exceed eight nurse' working hours. The time available is a problem restriction. The resources necessary for some of the tasks are food, equipment and rooms, among others. AGALZ takes care of incidents and interruptions that may occur during replanning (Bajo, Corchado & Castillo, 2006). Furthermore AGALZ trusts the nurse in the sense that the revision of a plan is made by the nurse. Finally, AGALZ learns about this new experience. If the evaluation of the plan is at least a 90% similar, the case is stored in the cases memory.

Case Study

A prototype of the system has been tested in several geriatric residences, which have been interested in improving the services offered to its patients and has collaborated in the development of the technology presented here, providing their know-how and experimenting with the prototype developed.

Figure 2 shows a basic schema of the wireless technology implemented in the residences. We selected 30 patients to test the system, so the hardware implemented basically consisted of 42 ID door readers, one on each door and elevator, 4 controllers, one at each exit, one

Figure 2. ALZ-MAS wireless technology organization

P
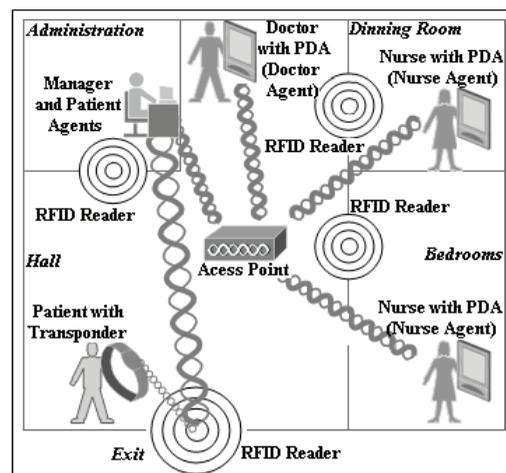
in the first floor hall and another in the second floor hall, and 36 bracelets, one for each patient and the nurses. The ID door readers get the ID number from the bracelets and send the data to the controllers which send a notification to the Manager agent, located in a central computer. To test the system 30 Patient Agents, 10 AGALZ Agents, 2 Doctor Agents and 1 Manager Agent were instantiated.

ALZ-MAS: Alzheimer Health Care Multi-Agent System

The characteristics of multi-agent systems make them appropriate for implementing into geriatric residences to improve health care of patients (Nealon & Moreno, 2003). A multi-agent system is a distributed system based on the cooperation of autonomous agents. The relationships established between the agents of ALZ-MAS are inspired in human's behaviours (doctors, nurses, patients, etc.) (Marrow, 2000).

Conclusions obtained after studying the requirements of the problem are that ALZ-MAS is composed of four different agent types as shown in Figure 3:

-Patient Agent manages the patient's personal data and behaviour (monitoring, location, daily tasks, and anomalies). Every hour validates the patient location, monitors the patient state and sends a copy of its memory base (patient state, goals and plans) to the Manager Agent in order to maintain backups. The patient state is instantiated at execution time as a set of beliefs and these beliefs are controlled through goals that must be achieved or maintained. The beliefs that were seen to define a general patient state at the Residences, were: weight, temperature, blood pressure, feeding, oral medication, parenteral medication, posture change, toileting, personal hygiene, and exercise. The beliefs and goals used for every patient depend on the plan (treatment) or plans that the doctors prescribe. Patient Agents monitors the patient state by means of the goals. It is necessary to maintain continuous communication with the rest of ALZ-MAS Agents, especially with AGALZ (through which the nurse can communicate the result of her assigned tasks). At least once per day, depending on the corresponding treatment, Patient Agents must communicate with AGALZ and Doctor Agents. Finally, Patient Agents must ensure that all actions indicated in the treatment are taken out. Patient Agents run on a central computer.

- Manager Agent plays two roles the security role that controls the patients' location and manages locks and alarms; and the Manager role that manages the medical record database and the doctor-patient and nurse-patient assignment. It must provide security for the patients and medical staff and the patients, doctors and nurse assignment must be efficient. This assignation is carried out through a CBR reasoning engine, which is incorporated within the Manager Agent. When a new assignation of tasks needs to be carried out to nurses or doctors, both past experiences, such as the profile of the nurse or doctor, and the

Figure 3. ALZ-MAS architecture: Doctor, AGALZ, Patient, and Manager Agents, within their interactions
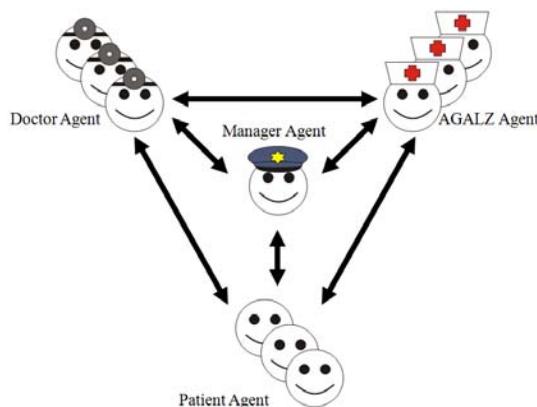
needs of the current situation are recalled. In this way, tasks are allocated to nurses. A nurse profile includes nurse's preferences such as holidays, etc.

Manager Agent runs on a central computer.

- Doctor Agent treats patients. It needs to interact with Patient Agents to order treatments and receive periodic reports, with the Manager Agent to consult medical records and assigned patients, and with AGALZ agents to ascertain patients' evolution.

- AGALZ schedules the nurse's working day obtaining dynamic plans depending on the tasks needed for each assigned patient. AGALZ manages nurses' profiles, tasks, available time and resources. The generated plans must guarantee that all the patients assigned to the nurse are given care. Nurses can't exceed 8 working hours. Every agent generates personalized plans depending on the nurse's profile and working habits. AGALZ Agents run on mobile devices, where each nurse can see her plans task by task. A plan can be interrupted for different reasons: a resource fails; a patient suffers a crisis and requires unforeseen attention; a patient has an unexpected visit; etc.

Extracting Results from ALZ-MAS

Figure 4 shows the average number of nurses working simultaneously (each of the 24 hours of the day) before and after the implantation of the system prototype into a test residence, with data collected for 6 months. The average number of patients was the same before and after the implementation. Tasks executed by nurses were divided in two categories: direct action tasks (where nurses are in contact with patients) and indirect action tasks (where nurses are not directly involved with patients, like monitoring, written reports, managing personal visits to the patients, etc.). During the first 3 months, the problem was analysed, the residence was observed and data was retrieved. Finally averages of the time spent by nurses in the carrying out of the tasks for every patient were obtained, having into account that a task depends on the dependency level of a patient and the nurse skill. For the direct action tasks, the following times were obtained for each patient: 35' cleaning, 18' feeding, 8' oral medication, 30' parenteral medication, 25' posture change, 8' toileting, 60' exercise and 10' others. We are especially interested on time spent on indirect action tasks; daily average times obtained for every kind of task before and after the implementation for each task can be seen on Table 1.

The system facilitates the more flexible assignation of the working shifts at the residence; since the workers have reduced the time spent on routine tasks and can assign this time to extra activities. Their work is automatically monitored, as well as the patients' activities. The stored information may be analysed with knowledge discovery techniques and may help to improve the quality of life for the patients and the efficiency of the centre (Marrow, 2000). The security

Figure 4. Number of nurses working simultaneously in the residence

P
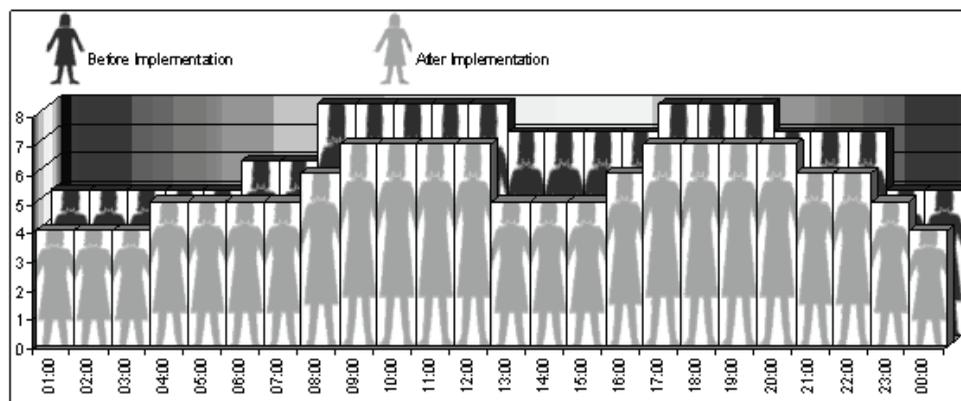
Table 1. Time (minutes) spent on indirect tasks

	Monitoring	Reports	Visits	Other	TOTAL
Before	167	48	73	82	370
After	105	40	45	60	250

of the centre has also been improved in two ways: the system monitors the patients and guarantees that each one of them is in the right place, and secondly, only authorised personnel can gain access to the residence protected areas.

FUTURE TRENDS

In the future, health care will require the use of new technologies that allow medical personnel to carry out their tasks more efficiently (Camarinha-Matos & Afsarmanesh, 2002). We are interested in the use of Ambient Intelligence (Ducatel, Bogdanowicz, Scapolo, Leijten & Burgelman), which provides a framework for the development of transparent, ubiquitous and unobtrusive environments. The objective of Ambient Intelligence is to adapt the existing technologies to the human necessities (Emiliani & Stephanidis, 2005). In this sense, the planner proposed in this work must be adapted to any other possible technologies an evaluated in similar environments.

CONCLUSION

We have shown the potential of deliberative AGALZ agents in a distributed multi-agent system focused on health care, providing a way to respond to some challenges of health care, related for example to the identification, control and health care planning. In addition, the use of RFID technology (Sokymat, 2006) on people provides a high level of interaction among users and patients through the system and is fundamental in the construction of an intelligent environment. Furthermore, the use of mobile devices, when used well, can facilitate social interactions and knowledge transfer.

REFERENCES

Aamodt, A. & Plaza, E. (1994). Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1):39-59, 1994.

Bajo, J., Corchado, J.M. & Castillo, L.F. (2006). Running Agents in mobile devices. *Lectures Notes in Computer Science*. Volume 4140 / 2006. Advances in Artificial Intelligence - IBERAMIA-SBIA 2006 October. Pp. 58-67 ISSN 0302-9743.

Bajo, J., Tapia, D.I., De Luis, A., Rodríguez, S. & Corchado, J.M. (2007). Nature-Inspired Planner Agent for Health Care. *Lecture Notes in Artificial Intelligence*. Volume 4507. Proceedings of IWANN'07. Pp. 1090-1097. ISSN: 0302-9743.

Bratman, M.E. (1987). Intentions, Plans and Practical Reason. *Harvard University Press*, Cambridge, M.A.

Camarinha-Matos, L.M. & Afsarmanesh, H. (2002). Design of a Virtual Community Infrastructure for Elderly Care. In *Proceedings of the IFIP Tc5/Wg5.5 Third Working Conference on infrastructures For Virtual Enterprises: Collaborative Business Ecosystems and Virtual Enterprises* (May 01 - 03, 2002). L. M. Camarinha-Matos, Ed. IFIP Conference Proceedings, vol. 213. Kluwer B.V., Deventer, The Netherlands, 635.

Corchado, J.M. & Laza, R. (2003). Constructing Deliberative Agents with Case-based Reasoning Technology. *International Journal of Intelligent Systems*. Vol. 18, No. 12, pp. 1227-1241. December, 2003.

Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J. & Burgelman, J.C. (2001). That's what friends are for. Ambient Intelligence (AmI) and the IS in 2010. *Innovations for an e-Society. Challenges for* Technology Assessment. Berlin, Germany.

Emiliani P.L. & Stephanidis, C. (2005). Universal access to ambient intelligence environments: opportunities and challenges for people with disabilities. *IBM Systems Journal*.

Glez-Bedia, M. & Corchado, J.M. (2002). A planning strategy based on variational calculus for deliberative agents. Computing and Information Systems Journal. Vol. 9(1), pp. 2-13. ISSN 1352-9404.

Marrow, P. (2000). Nature-Inspired computing technology and applications. *BT Technology Journal*. 18, 4. October 2000.

Nealon, J.L. & Moreno, A. (2003). Applications of Software Agent Technology in the Health Care Domain. *Whitestein Series in Software Agent Technologies*. Birkhäuser-Verlag, Basel, Germany.

Pokahr, A., Braubach, L. & Lamersdorf, W. (2003). Jadex: Implementing a BDI-Infrastructure for JADE Agents. In Search of Innovation. (3) 76-85.

Pokahr, A., Braubach, L. & Lamersdorf, W. (2003). Jadex: Implementing a BDI-Infrastructure for JADE Agents. In: *EXP - in search of innovation (Special Issue on JADE)*. pp. 76-85.

Sokymat. (2006). *ASSA ABLOY Identification Technologies*. http://sokymat.aaitg.com

Tapia, D.I., Bajo, J., Corchado, J.M., Rodríguez, S. & Manzano, J.M. (2007). Hybrid Agents Based Architecture on Automated Dynamic Environments. *Proceedings of Knowledge-Based Intelligent Information and Engineering Systems, 11th International Conference, KES 2007*. Lecture Notes in Computer Science 4693 Springer.

KEY TERMS

Ambient Intelligence (AmI): Refers to electronic environments that are sensitive and responsive to context and people needs and characteristics. It is characterized by systems and technologies that are embedded, context-aware, ubiquitous, non intrusive, personalized, adaptive and anticipatory.

Case-Based Reasoning: A type of reasoning based on the use of past experiences. The purpose of CBR systems is to solve new problems by adapting solutions that have been used to solve similar problems in the past. The main concept when working with CBR is the concept of case, which can be defined as a past experience.

Case-Based Planning: A specialization of Case-Based Reasoning in which the solution proposed by the system is a plan (a sequence of actions).

CBR-BDI: A deliberative BDI agent that incorporates a CBR motor as reasoning mechanism.

CBP-BDI: A deliberative BDI agent specialized in generating plans. It incorporates a Case-Based Planning mechanism.

Multi-Agent System: A system composed of several intelligent autonomous agents, collectively capable of reaching goals solving problems in a distributed way.

Radio Frequency Identification: A wireless technology used to identify and receive information on the move. An RFID system contains basically four components: tags, readers, antennas and software.

Privacy–Preserving Estimation

Mohammad Saad Al-Ahmadi
King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia

Rathindra Sarathy
Oklahoma State University, USA

INTRODUCTION

Data mining has evolved from a need to make sense of the enormous amounts of data generated by organizations. But data mining comes with its own cost, including possible threats to the confidentiality and privacy of individuals. This chapter presents a background on privacy-preserving data mining (PPDM) and the related field of statistical disclosure limitation (SDL). We then focus on privacy-preserving estimation (PPE) and the need for a data-centric approach (DCA) to PPDM. The chapter concludes by presenting some possible future trends.

BACKGROUND

The maturity of information, telecommunications, storage and database technologies, have facilitated the collection, transmission and storage of huge amounts of raw data, unimagined until a few years ago. For raw data to be utilized, they must be processed and transformed into information and knowledge that have added value, such as helping to accomplish tasks more effectively and efficiently. Data mining techniques and algorithms attempt to aid decision making by analyzing stored data to find useful patterns and to build decision-support models. These extracted patterns and models help to reduce the uncertainty in decision-making environments.

Frequently, data may have sensitive information about previously surveyed human subjects. This raises many questions about the privacy and confidentiality of individuals (Grupe, Kuechler, & Sweeney, 2002). Sometimes these concerns result in people refusing to share personal information, or worse, providing wrong data.

Many laws emphasize the importance of privacy and define the limits of legal uses of collected data. In the healthcare domain, for example, the U.S. Department of Health and Human Services (DHHS) added new standards and regulations to the Health Insurance Portability and Accountability Act of 1996 (HIPAA) to protect "*the privacy of certain individually identifiable health data*" (HIPAA, 2003). Grupe et al. (2002, Exhibit 1, p. 65) listed a dozen privacy-related legislative acts issued between 1970 and 2000 in the United States.

On the other hand, these acts and concerns limit, either legally and/or ethically, the releasing of datasets for legitimate research or to obtain competitive advantage in the business domain. Statistical offices face a dilemma of legal conflict or what can be called "war of acts." While they must protect the privacy of individuals in their datasets, they are also legally required to disseminate these datasets. The conflicting objectives of the Privacy Act of 1974 and the Freedom of Information Act is just one example of this dilemma (Fienberg, 1994). This has led to an evolution in the field of statistical disclosure limitation (SDL), also known as statistical disclosure control (SDC).

SDL methods attempt to find a balance between data utility (valid analytical results) and data security (privacy and confidentiality of individuals). In general, these methods try to either (a) limit the access to the values of sensitive attributes (mainly at the individual level), or (b) mask the values of confidential attributes in datasets while maintaining the general statistical characteristics of the datasets (such as mean, standard deviation, and covariance matrix). *Data perturbation* methods for microdata are one class of masking methods (Willenborg & Waal, 2001).

Data Mining vs. Statistical Analysis

Statisticians and researchers conduct surveys and collect datasets that are considered to be large when they contain a few hundred records (Hand, 1998). Traditional statistical techniques are the main (and the most suit-

able) tools for analyzing these datasets to make inferences and estimate population parameters. When the size of datasets is large, traditional statistical analysis techniques may not be the appropriate tools (Hand, 1998, 2000; Hand, Blunt, Kelly, & Adams, 2000). First, traditional statistical analysis may be inappropriate because almost any small difference in a large dataset becomes statistically significant. Second, large datasets may suggest that data was not collected for inference (parameter estimation) about the population. Third, in businesses, a significant amount of data is generated because of unplanned activities (e.g., transactional databases) and not from planned activities (e.g., experiment or survey designs). Therefore, for large datasets, data mining becomes more appropriate.

Examples of large datasets are abundant. Market-Touch, a company located in Georgia, USA, supports direct marketers with data and analytical tools (DMReview.com, 2004). It has a six-terabyte database called Real America Database (RADBÒ), which provides information about more than 93 million households and 200 million individuals. It is updated monthly with more than 20 million records.

Statistical agencies also experience this phenomenon of rapidly growing datasets. The US Census Bureau (Census, 2001) reported that the Census 2000 data consist of "*information about the 115.9 million housing units and 281.4 million people across the United States.*" These large sizes suggest the need for analytical tools that are suitable for large datasets, and again, data mining tools naturally come into play. Consequently, the Bureau provides programs with data mining capabilities such as DataFerrett (Federated Electronic Research, Review, Extraction and Tabulation Tool), which can be used to analyze and extract data from TheDataWeb - a repository of datasets that cover more than 95 subject areas.

Motivation for Privacy-Preserving Data Mining (PPDM)

Data mining techniques may lead to more significant threats to privacy and confidentiality than statistical analysis. Domingo-Ferrer and Torra (2003) make a connection between SDL methods and some data-mining AI (artificial intelligence) tools and suggest that disclosure and re-identification threats can be magnified.

DM tools can be used to aggregate or combine masked copies of a specific original dataset to reverse masking and re-build the original dataset, which raises a *confidentiality* issue. This is particularly true when unsophisticated SDL techniques are used and many masked copies are released. DM tools can also be used to enforce data integrity and consistency in distributed datasets by re-identifying different records belonging to the same individual raising a *privacy* issue.

These concerns about privacy and confidentiality when DM tools are used have led to the birth of privacy-preserving data mining (PPDM). The main goal of PPDM is to find useful patterns and build accurate models from datasets without accessing the individuals' precise original values in records of datasets (Agrawal & Srikant, 2000).

Related Work in Privacy-Preserving Data Mining (PPDM)

Similar to the classification of data mining (DM) techniques proposed by Berry and Linoff (2004), privacy-preserving data mining (PPDM) techniques can be classified as: (a) directed PPDM techniques: privacy-preserving estimation and privacy-preserving classification, and (b) undirected PPDM techniques: privacy-preserving association rules and privacy-preserving clustering.

Directed PPDM techniques try to model the relationship between a dependent variable and other (independent) variables in masked datasets. *Estimation* deals with continuous dependent variables and *classification* with categorical or binary dependent variables. The models obtained from the masked data using directed PPDM techniques must be the same (or similar) to that from the original dataset at the aggregate level, while protecting the privacy and confidentiality at the individual level.

In undirected PPDM, there is no concept of a dependent variable. Instead, the goal is to find unknown patterns and rules. *Clustering* is used to discover (and usually profile) homogenous subsets of data records and often used as a preprocessing tool (to segment the customer base, for example) before applying other DM technique (Berry & Linoff, 2004). *Association rules* are used to discover which items go together (are associated). Again, the goal of PPDM is to obtain similar

Figure 1. Privacy-preserving data mining PPDM literature

patterns from both the masked and original data. Figure 1 , reproduced from Al-Ahmadi (2006), shows an abstract view of privacy-privacy data mining (PPDM) literature broken down by technique. Details on the references may be found in Al-Ahmadi (2006).

PRIVACY-PRESERVING ESTIMATION (PPE)

We focus on privacy-preserving estimation (PPE) (also called privacy-preserving regression). PPE is still in its infancy compared to other PPDM methods, with some approaches showing promise. Sanil et al. (2004) proposed an algorithm for computing the exact coefficients of multiple linear regression for *vertically*-distributed (or partitioned) dataset without sharing original values. The dataset is assumed to contain a single shared, non-confidential dependent variable. The unshared confidential, independent variables are owned by more than two parties (agents) involved in the estimation

process. It utilizes the secure summation algorithm (Benaloh, 1987; Clifton, Kantarcioglu, Vaidya, Lin, & Zhu, 2002) to share a statistical summary (total), populated partially by each party without revealing how much each party contributes to that statistic. This total is needed for estimating the regression coefficients iteratively. Thus, each party can calculate accurately, the coefficients of the variables they own and share them with other parties.

Karr et al. (2004) suggest two approaches for building multiple linear regression on the union of a *horizontally*-distributed dataset. The first approach, (*secure data integration*) integrates horizontally-distributed datasets from multiple parties (agents) into one dataset, while protecting the identity of the data source. Each party could locally run linear regression analysis on the integrated dataset. This approach only protects the identity of the data sources (i.e. the identity of the involved parties, not the identity or confidentiality of surveyed human subjects). A second approach is based on the additive nature of the linear regression analysis.,

Statistics (rather than data) needed to calculate the least squares estimators of linear regression coefficients are shared and integrated in a secure manner using the secure summation algorithm (Benaloh, 1987; Clifton et al., 2002; Schneier, 1996).

Remote regression servers (cf. Duncan & Mukherjee, 2000; Keller-McNulty & Unger, 1998; Schouten & Cigrang, 2003) are access-limitation (not masking) methods for protecting microdata for building linear regression models. Although this approach builds linear regression models using original values, users do not usually have any means of checking the fit of their models. Reiter (2003) proposed a method to overcome this limitation based on releasing artificial, simulated (marginally-wise) dependent and independent variables, residuals and fitted values that mimic the original relationships of the built models.

Because many multivariate methods, including multivariate linear regression, depend on matrix computations such as matrix multiplication and matrix inverse, Du et al. (2004) proposed secure two-party matrix computations protocols. These enable two agents to collaboratively run matrix computations without knowing or accessing the other party's original, sensitive values, and without the involvement of a third party.

The above approaches to PPE, they are focused exclusively on linear relationships. This makes them somewhat limited for more general purpose PPE, where nonlinear relationships found in the original data may need to be preserved in the masked data.

Data-Centric Approach (DCA) for Privacy-Preserving Data Mining

One of the problems with many existing PPDM approaches is that they create a dependency between the algorithm and the dataset (Thuraisingham, 2005); see, for example, Agrawal and Srikant (2000). The PPDM algorithm is usually a modification of a specific DM algorithm, for a specific protection technique. The masked data can therefore be analyzed using only that particular (tailored) data mining algorithm. Otherwise there is no guarantee that the results from analyzing the masked dataset will be the same as, or similar to, that from analyzing the original dataset. This is not a good idea because data miners usually employ more than one algorithm to mine a dataset. Examining all data mining algorithms, as well as modifying them, is not feasible.

Second, once a dataset is released, there is no guarantee as to which algorithm might be applied possibly leading to incorrect conclusions and actions.

Instead, as suggested by Al-Ahmadi et al. (2004), datasets should be protected or masked without reference to a specific DM algorithm. Oliveira and Zaïane (2004b) support the concept of a Data-Centric Approach (DCA) which supports the concept that the masking algorithm must *not* be tied to the data mining algorithm, but must be based on the characteristics of the dataset and its subsequent use. For example, a good PPE algorithm will mask the dataset based on the kind of relationships that need to be maintained in the masked dataset. However, it will not mandate that a particular data mining algorithm should be used to perform the estimation using the masked data. Al-Ahmadi (2006) demonstrates some PPE algorithms that utilize the DCA approach. Oliveira and Zaïane (2004a) also applied the DCA concept by developing a new PPDM clustering algorithm called Rotation-Based Transformation (RBT) that allows any distance-based clustering algorithms to be used on the masked datasets.

FUTURE TRENDS

Data perturbation and SDL masking methods can be a good starting point for implementing DCA in PPE and PPDM. One protection method used is Simple Additive Data Perturbation Method (SADP) (Traub, Yemini, & Wozniakowski, 1984), which has undesirable characteristics in terms of data utility and data security (Muralidhar, Parsa, & Sarathy, 1999). Most of the newer and more sophisticated data perturbation and SDL masking methods, such as C-GADP (Sarathy, Muralidhar, & Parsa, 2002), IPSO (Burridge, 2003), EGADP (Muralidhar & Sarathy, 2005) and data shuffling (Muralidhar & Sarathy, 2003, 2006), have not been investigated in the PPE and the general PPDM domain. The only exception is the GADP method (Muralidhar et al., 1999), which appears in a few privacy-preserving classification studies (Islam & Brankovic, 2004; Wilson & Rosen, 2002, 2003; Wilson, Rosen, & Al-Ahmadi, 2005a, 2005b). Hence, there is a need to investigate the possibilities of using some of these advanced SDL masking methods in PPE and PPDM.

From another perspective, different types of relationships can exist in a dataset. For instance, multivariate

normal datasets guarantee that all existing relationships among variables are linear. For this special case, some existing SDL masking methods are readily available and can perfectly preserve linear relationships. This is due to the fact that most SDL methods are developed to preserve linear relationships. However, most (business) datasets contain nonlinear relationships (Zhang, 2004), which can be monotonic or non-monotonic (Fisher, 1970). "*A truth about data mining not widely discussed is that the relationships in data the miner seeks are either very easy to characterize, or very, very hard,*" (Pyle, 2003, p. 314). Therefore, there is a need to develop masking methods for PPE and PPDM to maintain more complicated types or relationships (i.e. monotonic nonlinear and non-monotonic relationships).

CONCLUSION

This article introduced privacy-preserving data mining (PPDM) and related concepts. It gave a brief overview of the four main PPDM techniques: estimation, classification, clustering, and association rules. Then, it reviewed some of the work that has been done in Privacy-Preserving Estimation (PPE). It concluded by discussing some of the possible future trends in PPDM and PPE including the need for research into data-centric SDL-based masking techniques for solving complicated PPE problems.

ACKNOWLEDGMENT

REFERENCES

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. *Sigmod Record, 29*(2), 439-450.

Al-Ahmadi, M., Sarathy, R., & Delen, D. (2004). *Privacy Preserving Data Mining: Issues and Opportunities.* Paper presented at the Workshop on Data Mining Research in Oklahoma, February 6, 2004, Tulsa, OK, USA.

Al-Ahmadi, M. S. (2006). *Adapting masking techniques for estimation problems involving non-monotonic relationships in privacy-preserving data mining.* Oklahoma State University, Stillwater, Oklahoma, USA.

Benaloh, J. C. (1987). Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret (Extended Abstract). Retrieved Jan 7, 2005, from http://research.microsoft.com/copyright/accept.asp?path=/crypto/papers/ssh.ps&pub=15.

Berry, M. J. A., & Linoff, G. (2004). *Data mining techniques: for marketing, sales, and customer relationship management* (2nd ed.). Indianapolis, Ind.: Wiley.

Burridge, J. (2003). Information preserving statistical obfuscation. *Statistics and Computing, 13*(4), 321-327.

Census. (2001). Introduction to Census 2000 Data Products. Retrieved Aug 2004, from http://www.census.gov/prod/2001pubs/mso-01icdp.pdf.

Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., & Zhu, M. Y. (2002). Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations Newsletter, 4*(2), 28-34.

DMReview.com. (2004). Industry Implementations [Electronic Version]. *Online News published in DM Direct Newsletter* from http://www.dmreview.com/article_sub.cfm?articleID=1004813.

Domingo-Ferrer, J., & Torra, V. (2003). On the connections between statistical disclosure control for microdata and some artificial intelligence tools. *Information Sciences, 151*(1), 153-170.

Du, W., Han, Y. S., & Chen, S. (2004). Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification. Retrieved Nov 21, 2004, from http://www.cis.syr.edu/~wedu/Research/paper/sdm2004_privacy.pdf.

Duncan, G. T., & Mukherjee, S. (2000). Optimal disclosure limitation strategy in statistical databases: Deterring tracker attacks through additive noise. *Journal of the American Statistical Association, 95*(451), 720-729.

Fienberg, S. E. (1994). Conflicts between the needs for access to statistical information and demands for

confidentiality. *Journal of Official Statistics, 10*(2), 115-132.

Fisher, R. A. (1970). *Statistical methods for research workers* (14th ed.). Darien, Conn.,: Hafner Pub. Co.

Grupe, F. H., Kuechler, W., & Sweeney, S. (2002). Dealing with data privacy protection: An issue for the 21st century. *Information Systems Management, 19*(4), 61-70.

Hand, D. J. (1998). Data mining: Statistics and more? *AMERICAN STATISTICIAN, 52*(2), 112-118.

Hand, D. J. (2000). Data mining: New challenges for statisticians. *Social Science Computer Review, 18*(4), 442-449.

Hand, D. J., Blunt, G., Kelly, M., & Adams, N. (2000). Data mining for fun and profit. *STATISTICAL SCIENCE, 15*(2), 111-126.

HIPAA. (2003). HIPAA Privacy Rule and Public Health - Guidance from CDC and the U.S. Department of Health and Human Services [Electronic Version] from http://www.cdc.gov/mmwr/preview/mmwrhtml/m2e411a1.htm.

Islam, M. Z., & Brankovic, L. (2004). *A framework for privacy preserving classification in data mining* Paper presented at the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation Dunedin, New Zealand.

Karr, A. F., Lin, X., Sanil, A. P., & Reiter, J. P. (2004). Secure Regression on Distributed Databases. Retrieved Nov 10, 2004, from http://www.niss.org/technicalreports/tr141.pdf.

Keller-McNulty, S., & Unger, E. A. (1998). A Database System Prototype for Remote Access to Information Based on Confidential Data. *Journal of Official Statistics, 14*(4), 347-360.

Muralidhar, K., Parsa, R., & Sarathy, R. (1999). A general additive data perturbation method for database security. *Management Science, 45*(10), 1399-1415.

Muralidhar, K., & Sarathy, R. (2003). *The Data Shuffle: A New Masking Procedure for Numerical Data.* Paper presented at the 8th INFORMS Computing Society, Chandler, AZ.

Muralidhar, K., & Sarathy, R. (2005). An Enhanced Data Perturbation Approach for Small Data Sets. *Decision Sciences, 36*(3), 513-529.

Muralidhar, K., & Sarathy, R. (2006). Data shuffling - A new masking approach for numerical data. *Management Science, 52*(5), 658-670.

Oliveira, S. R. M., & Zaïane, O. R. (2004a). *Achieving Privacy Preservation When Sharing Data For Clustering.* Paper presented at the International Workshop on Secure Data Management in a Connected World (SDM'04) in conjunction with VLDB 2004, August 2004, Toronto, Canada.

Oliveira, S. R. M., & Zaïane, O. R. (2004b). *Toward Standardization in Privacy-Preserving Data Mining.* Paper presented at the ACM SIGKDD 3rd Workshop on Data Mining Standards (DM-SSP 2004), August 22, 2004, Seattle, WA, USA.

Pyle, D. (2003). *Business modeling and data mining.* Amsterdam; Boston: Morgan Kaufmann Publishers.

Reiter, J. P. (2003). Model diagnostics for remote access regression servers. *Statistics and Computing, 13*(4), 371-380.

Sanil, A. P., Karr, A. F., Lin, X., & Reiter, J. P. (2004, August 22-25, 2004). *Privacy preserving regression modelling via distributed computation.* Paper presented at the 2004 ACM SIGKDD international conference on Knowledge discovery and data Seattle, WA, USA.

Sarathy, R., Muralidhar, K., & Parsa, R. (2002). Perturbing Non-normal confidential attributes: The copula approach. *Management Science, 48*(12), 1613-1627.

Schneier, B. (1996). *Applied cryptography : protocols, algorithms, and source code in C* (2nd ed.). New York: Wiley.

Schouten, B., & Cigrang, M. (2003). Remote access systems for statistical analysis of microdata. *Statistics and Computing, 13*(4), 381-389.

Thuraisingham, B. (2005). Privacy-Preserving Data Mining: Development and Directions. *Journal of Database Management, 16*(1), 75-87.

Traub, J. F., Yemini, Y., & Wozniakowski, H. (1984). The Statistical Security of a Statistical Database. *ACM Transactions on Database Systems, 9*(4), 672-679.

P

Willenborg, L. C. R. J., & Waal, T. d. (2001). *Elements of statistical disclosure control*. New York: Springer.

Wilson, R. L., & Rosen, P. A. (2002). *The Impact of Data Perturbation Techniques on Data Mining Accuracy.* Paper presented at the 33rd Annual Meeting of the Decision Sciences Institute.

Wilson, R. L., & Rosen, P. A. (2003). Protecting Data through Perturbation Techniques: The Impact on Knowledge Discovery in Databases. *Journal of Database Management, 14*(2), 14-26.

Wilson, R. L., Rosen, P. A., & Al-Ahmadi, M. S. (2005a). Knowledge Structure and Data Mining Techniques. In D. G. Schwartz (Ed.), *Encyclopedia of knowledge management* (pp. 523-529). Hershey, PA: Idea Group Reference.

Wilson, R. L., Rosen, P. A., & Al-Ahmadi, M. S. (2005b). Secure Knowledge Discovery in Databases. In D. G. Schwartz (Ed.), *Encyclopedia of knowledge management* (pp. 787-794). Hershey, PA: Idea Group Reference.

Zhang, G. P. (Ed.). (2004). *Neural networks in business forecasting*. Hershey, Pa.: Idea Group.

KEY TERMS

Confidentiality: The status accorded to specific attributes (such as salary) in datasets, whose original values should not be revealed. Generally, some type of protection such as masking must be provided before these confidential attributes are disseminated.

Data Mining Algorithm: A systematic, practical method to implement a data mining *technique*. Different algorithms can be used to implement the same data mining *technique*. For example, decision trees algorithms (CART, C4.5, C5, etc.) and logistic regression are among the algorithms of the *classification* data mining *technique*.

Data Mining Technique: The main purpose or objective of the data mining modelling process. Each technique can be implemented using different DM *algorithms*.

Data-Centric Approach (DCA): The concept that data protection techniques must be independent of (standard) DM algorithms. That is, the masked data must be analyzable using multiple DM algorithms while providing results comparable to the results from analyzing the original data.

Privacy: Privacy is the desire of individuals to control their personal information. Generally, in the SDL literature, it relates to the identity of an individual, while confidentiality relates to specific information about the individual (such as salary).

Statistical Disclosure Limitation (SDL) or Statistical Disclosure Control (SDC): A set of methods that attempt to protect privacy and confidentiality of data, while preserving the overall statistical characteristics of original datasets (such as mean and covariance matrix) in the protected dataset.

Protein Structure Prediction by Fusion, Bayesian Methods

Somasheker Akkaladevi
Virginia State University, USA

Ajay K. Katangur
Texas A&M University – Corpus Christi, USA

Xin Luo
The University of New Mexico, USA

INTRODUCTION

Prediction of protein secondary structure (alpha-helix, beta-sheet, coil) from primary sequence of amino acids is a very challenging and difficult task, and the problem has been approached from several angles. A protein is a sequence of amino acid residues and can thus be considered as a one dimensional chain of 'beads' where each bead correspond to one of the 20 different amino acid residues known to occur in proteins. The length of most protein sequence ranges from 50 residues to about 1000 residues but longer proteins are also known, e.g. myosin, the major protein of muscle fibers, consists of 1800 residues (Altschul et al. 1997). Many techniques were used many researchers to predict the protein secondary structure, but the most commonly used technique for protein secondary structure prediction is the neural network (Qian et al. 1988).

This chapter discusses a new method combining profile-based neural networks (Rost et al. 1993b), Simulated Annealing (SA) (Akkaladevi et al. 2005; Simons et al. 1997), Genetic algorithm (GA) (Akkaladevi et al. 2005) and the decision fusion algorithms (Akkaladevi et al. 2005). Researchers used the neural network (Hopfield 1982) combined with GA and SA algorithms, and then applied the two decision fusion methods; committee method and the correlation methods and obtained improved results on the prediction accuracy (Akkaladevi et al. 2005). Sequence profiles of amino acids are fed as input to the profile-based neural network. The two decision fusion methods improved the prediction accuracy, but noticeably one method worked better in some cases and the other method for some other sequence profiles of amino acids as input (Akkaladevi et al. 2005). Instead of compromising on some of the good solutions that could have generated from either approach, a combination of these two approaches is used for obtaining better prediction accuracy. This criterion is the basis for the Bayesian inference method (Anandalingam et al. 1989; Schmidler et al. 2000; Simons et al. 1997). The results obtained show that the prediction accuracy improves by more than 2% using the combination of the decision fusion approach and the Bayesian inference method.

BACKGROUND

A lot of interesting work has been done on protein secondary structure prediction problem, and over the last 10 to 20 years the methods have gradually improved in accuracy. The most successful application of neural networks (Hopfield 1982) to secondary structure prediction was obtained by Rost and Sander (Rost et al. 1993b; Rost et al. 1993c; Rost 1996; Rost et al. 1994), which resulted in the prediction mail server called PHD (Rost et al. 1993c). Using profile-based neural network and a few other methods, the performance of the network is reported to be up to 67.2% (Rost et al. 1993b).

In the problem of the protein secondary structure prediction, the inputs are the amino acid sequence profiles while the output is the predicted structure (also called conformation, which is the combination of alpha helices, beta sheets and loops) (Banavar et al. 2001; Branden et al. 1999). A typical protein sequence and its conformation class are shown below:

Protein Sequence: ADADADADCCQQFFFAAAQQA-
    QQA
Conformation Class: HHHH EEEE HHHHHHHH

P

H stands for Helical, E for Extended, and blanks are the remaining coiled conformations.

A typical protein contains about 32% alpha helices, 21% beta sheets and 47% loops or non-regular structure (Rost et al. 1993b). It is possible to predict loop regions with higher accuracy than alpha helices or beta sheets (Rost et al. 1993c). The *seven-fold cross-validation* technique is used on the set of 126 non-homologous globular proteins from (Rost & Sander, 1994), which is called the RS126 data set (Rost et al. 1994) for training and testing purpose.

The protein secondary structure accuracy is calculated by using the three-state per-residue accuracy (Q_3), which gives the percentage of correctly predicted residues in either of the three states (classes), alpha helix, beta strand or loop region (Qian et al. 1988; Rost 1996):

$$Q_3 = \left[\frac{(P_\alpha + P_\beta + P_{loop})}{T} \right] \times 100\%$$

P_α, P_β and P_{loop} are number of residues predicted correctly in state alpha helix, beta strand and loop respectively while T is the total number of residues.

PROTEIN SECONDARY STRUCTURE PREDICTION BY VARIOUS APPROACHES

In this research the RS126 dataset is used, which contains 126 sequences with approximately more than 23,300 amino acid positions and 20 amino acids (Rost et al. 1994). Orthogonal encoding scheme is used for the input which is sent to the profile-based neural network.

Protein Secondary Structure Prediction using sequence profiles **-** The profile-based neural network is used for this research. Using profiles at the input level generally has been shown to yield better results than using profiles at the output level (Baldi et al. 1999; Rost et al. 1993b). Using this approach the secondary structure prediction accuracy (Q_3) is 66.8%.

GA and the profile-based Neural Networks for protein secondary structure prediction **-** The predicted structure from the profile-based neural network is given to GA; the GA does a series of mutation and crossover operations on the predicted structure from

the profile-based neural network to generate new solutions (offspring's) (Akkaladevi et al. 2005). After the offspring is generated; the fitness of this new offspring is calculated by again comparing to the true structure already known by using the Q_3 function. The GA accepts or rejects this solution depending on the fitness value, which in this case is the prediction accuracy Q_3. Finally at this point the error value is calculated and back-propagated to adjust the weights of the profile-based neural network. The mutation probability for GA in this research is set at 0.25, number of generation's at 75, population size at 30 and the crossover probability as 100% (Akkaladevi et al. 2005). Using this approach the secondary structure prediction accuracy (Q_3) is 69.2%.

SA and the profile-based Neural Networks for protein secondary structure prediction **-** The predicted structure from the profile-based neural network is sent to the SA algorithm for further processing by the SA algorithm (Akkaladevi et al. 2005). The SA algorithm generates new solutions and compares it with the true secondary structure which is already known to calculate the prediction accuracy Q_3. The error is than calculated by determining the value of Q_3. This error value is then back-propagated to adjust the weights of the profile-based neural network. The starting temperature for SA in this research is set at 600, the final temperature at 0.20, the temperature cooling rate at 0.84, and the number of iterations per temperature at 20 (Akkaladevi et al. 2005). Using this approach the secondary structure prediction accuracy (Q_3) is 68.3%.

Prediction of protein secondary structure using the Committee method and the profile-based Neural Network **-** In the committee based method (Mazurov et al. 1987) of applying decision fusion the secondary structure values are calculated using a combined profile-based neural network (PNN) with GA, a combined profile-based neural network with SA, and the independent profile-based neural network. The output obtained from the profile-based neural network, combined profile-based neural network plus GA and combined profile-based neural network plus SA is routed to the decision fusion algorithm, for fusing the solutions as shown in Figure 1 (Akkaladevi et al. 2005).

The decision fusion (Abidi et al. 1992) algorithm works on the basis of a committee (committee method or voting method), where each individual in the committee decides on the best solution according to pre-determined rules and then cast their vote for the

Figure 1. Fusing the various solutions according to the fusion rules
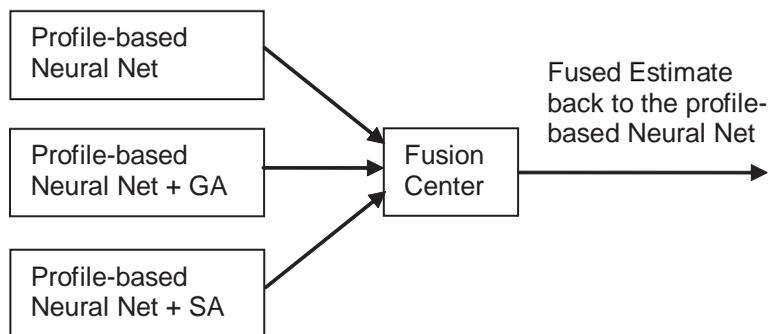
best approach (Mazurov et al. 1987). In the event of a tie, the tie is broken by one more rule; the priority assigned to each algorithm. The algorithm with the highest priority wins. The Committee fusion algorithm is outlined below:

1. Given a secondary structure output obtained by profile-based neural network of N_i elements, where $i = 1,2,……,n$. (Here for 'H' we assume a value of 2, for 'E' a value of 3, and for 'C' a value of 4. These are arbitrarily chosen values). Similarly represent output from GA and SA by G_i and S_i respectively

2. Calculate the following values:

$$G = \sum_{i=1}^{n}(N_i - G_i)^2 \tag{1}$$

$$S = \sum_{i=1}^{n}(N_i - S_i)^2 \tag{2}$$

$$N = 0 \tag{3}$$

3. Compute $N_i - G_i$. If $N_i - G_i > 0$, then (bin+) $\leftarrow N_i - G_i$ else if $N_i - G_i < 0$, then (bin-) $\leftarrow N_i - G_i$, where bin+ and bin- are the so called positive and negative bins. If the result of the operation is zero, it is not stored in any of the bins.

4. Evaluate bin+ and bin-, the positive and negative bins for G; if they are equal or if the positive bin has a higher count compared to the negative bin G is assigned a positive sign ($+G$), else G is

assigned a negative sign ($-G$). Always consider $N=0$.

5. Repeat steps 3 and 4 to calculate S.

6. Use $max(N, G, S)$ to be the secondary structure for calculating Q_3 which is used to determine the error for back-propagation for weight adjustments. Each algorithm votes for the best solution by comparing its value with the other algorithms values. The algorithm with the majority votes wins the race. In the event of a tie, the tie is broken according to the algorithm's priority, and the algorithm which wins calculates the prediction accuracy using the function Q_3 to determine the error that is to be back-propagated to the profile-based neural network for weight adjustments.

7. The profile-based neural network (PNN) secondary structure values are assigned the highest priority, followed by the combination of profile-based neural network and GA (PNN+GA), and then followed by the combination of profile-based neural network and SA (PNN+SA) (Akkaladevi et al. 2005). Using this approach the secondary structure prediction accuracy (Q_3) is 70.8%.

Prediction of protein secondary structure using the Correlation method and the profile-based Neural Network - This method is very similar to the committee method but with some minor changes (Akkaladevi et al. 2005; Ho et al. 1994). In this method the algorithm that wins after decision fusion is applied is used to calculate the prediction accuracy using the function Q_3 to determine the error that is to be back-propagated to the profile-based neural network for weight adjustments.

After this adjustment of weights on the profile-based neural network, the previous protein sequence is again used for testing purpose to check whether better prediction accuracy is achieved or not. Here the new weights are used if we get an improvement of more than 1.5%, otherwise from the previously calculated prediction accuracies of (PNN), (PNN+GA) and (PNN+SA), the method which produces the highest prediction accuracy is chosen to determine the error that is to be back-propagated to the profile-based neural network for weight adjustments (Akkaladevi et al. 2005). Using this approach the secondary structure prediction accuracy (Q_3) is 71.4%.

PREDICTION OF PROTEIN SECONDARY STRUCTURE BY THE BAYESIAN INFERENCE METHOD

In this method the Bayesian inference method is applied on the output generated by the committee and correlation methods of decision fusion (Anandalingam et al. 1989; Schmidler et al. 2000). In the Bayesian inference approach both these methods are used by assigning a specific probability value to them, and then generating a new value using the Bayesian equation (Anandalingam et al. 1989; Simons et al. 1997). This new value obtained is used to decide between the two methods (committee method and correlation method) to be used for calculating the error that is to be back-propagated to the profile-based neural network for weight adjustments. The following Bayesian equation is used to calculate the value for judging between the two methods (Anandalingam et al. 1989).

$$P(H_1 \mid D) = \frac{P(H_1) \times P(D \mid H_1)}{P(H_1) \times P(D \mid H_1) + P(H_2) \times P(D \mid H_2)}$$

To illustrate, let H_1 corresponds to correlation method, and H_2 corresponds to committee method. Since the correlation method produces better prediction accuracy compared to the committee method, for our first instance we assume that $P(H_1) = 0.51$, and $P(H_2) = 0.49$ (assigning more probability for choosing correlation method as this method produces better prediction accuracy compared to the committee method).

For example if we obtain a prediction accuracy of 71% using the correlation method and a prediction

accuracy of 70.5% using the committee method, then $P(D|H_1) = 0.71$ and $P(D|H_2) = 0.705$. Bayesian equation then yields:

$$P = \frac{0.51 \times 0.71}{0.51 \times 0.71 + 0.49 \times 0.705} = 0.5117$$

If the probability obtained is greater than or equal to 0.5, the correlation method is used for calculating the error that is to be back-propagated to the profile-based neural network for weight adjustments.

For example if we obtain a prediction accuracy of 69% using the correlation method and a prediction accuracy of 72% using the committee method, then $P(D|H_1) = 0.69$ and $P(D|H_2) = 0.72$. Bayesian equation then yields:

$$P = \frac{0.51 \times 0.69}{0.51 \times 0.69 + 0.49 \times 0.72} = 49.93$$

If the probability obtained is less than 0.5, the committee method is used for calculating the error to be back-propagated for weight adjustments.

Similarly this new approach is tested using various values of probability for $P(H_1)$ and $P(H_2)$, and always choosing $P(H_1)$ greater than $P(H_2)$. From the several test cases, it is concluded that the values of 0.506 for $P(H_1)$ and 0.494 for $P(H_2)$ produce the greatest prediction accuracy. Using the Bayesian approach the prediction accuracy is obtained to be 73.3% (Q_3). This method produces the highest protein secondary structure prediction accuracy compared to all the other methods investigated in this research.

SIMULATION RESULTS

The simulations are performed using code written in JAVA on a 3.6 GHz Intel Pentium IV PC with hyper-threading running Microsoft Windows XP with 2GB of RAM and a 160GB hard disk. The multi-threading approach is used for running the GA and SA algorithms and the decision fusion methods in parallel. Table 1 provides the summary of the prediction accuracies achieved using various approaches in this research.

It is clearly evident from Table 1 that the Bayesian inference method improves the prediction accuracy by 2% compared to that of correlation method and

Table 1. Comparison of prediction accuracy (Q$_3$) for various approaches

Approach Used	Prediction Accuracy (Q$_3$)
Profile-based Neural Network	66.8%
Profile-based Neural Network & GA	69.2%
Profile-based Neural Network & SA	68.3%
Decision fusion (Committee method) using Profile-based Neural Network	70.8%
Decision fusion (Correlation method) using Profile-based Neural Network	71.4%
Bayesian Inference method	73.3%

overall a prediction accuracy of 6.5% more than the profile-based neural network, which is a significant achievement.

FUTURE TRENDS

Many researchers all over the world are actively working on this problem using various methods to achieve at better prediction accuracy.

The future work can comprise the use of other decision fusion methods such as the clustering method, the fuzzy set method, and the probabilistic method for further improving on the protein secondary structure prediction accuracy.

CONCLUSION

This research aimed at improving the protein secondary structure prediction accuracy using the Bayesian inference method. Although there exists a variety of protein structure classification algorithms, research was performed in the belief that further improvement can be attained by finding the best way to combine several methods to lead to a unified better decision. From the results obtained we can conclude that applying AI algorithms along with decision fusion techniques improves the prediction accuracy compared to that of prediction by neural networks or AI algorithms individually or combined with profile-based neural networks. The simulations results prove that the Bayesian Inference method improves the prediction accuracy over the other decision fusion methods. The main advantage

of using this approach is that, it does not comprise the advantages provided by either committee or correlation methods of decision fusion.

REFERENCES

M. A. Abidi and R. C.Gonzales, eds. (1992). *Data Fusion in Robotics and Machine Intelligence*. Academic Press Inc.

Somasheker Akkaladevi, Ajay K Katangur, Saeid Belkasim, and Yi Pan. (2005). Protein Secondary Structure Prediction using decision fusion of Genetic Algorithm and Simulated Annealing Algorithm, *International Conference on Neural Networks and Brain*, Vol. 1, pp. 467-472, Beijing, China.

Altschul S.F., Madden T.L., Schaffer A.A., Zhang J., Zhang Z., Miller W., Lipman D.J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acid Research*, 25:3389-3402.

G.Anandalingam and L. Chen. (1989). Linear combination of forecasts: a general bayesian model, *Journal of Forecasting*, vol. 8, pp. 199–214.

Baldi P, Brunak S, Frasconi P, Pollastri G, Soda G. (1999). Exploiting the past and the future in protein secondary structure prediction, *Bioinformatics*; 15:937–946.

Banavar J.R and Maritan A. (2001). Computational Approach to the Protein-Folding Problem, *Proteins: Structure, Function, and Genetics*, 42: 433-435.

P

Branden, C. and Tooze, J. (1999). *Introduction to Protein Structure*, Garland Publishing.

T. K. Ho, J. J. Hull, and S. N. Srihari. (1994). Decision combination in multiple classifier systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 66– 75.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational properties, *Proceedings of the National Academy of Sciences of the USA*, 79:2554 -- 2588.

V.D.Mazurov, A.I.Krivonogov, and V.S.Kazantsev. (1987). Solving of optimization and identification problems by the committee methods, *Pattern Recognition*, vol. 4, no. 20, pp. 371–378.

Qian, N. and Sejnowski, T. (1988). Predicting the secondary structure of globular proteins using neural network models, *Journal of Molecular Biology*, 202:865-884.

Rost, B. and Sander, C. (1993b). Improved prediction of protein secondary structure by use of sequence structure and neural networks, *Proceedings of the National Academy of Sciences of the United States of America*, 90:7558-7562.

Rost, B. and Sander, C. (1993c). Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, 232:584-599.

Rost, B. (1996). Predicting 1d protein structure by profile based neural networks, *Meth. in Enzym.*, 266:525-539.

Rost, B., Sander, C., and Schneider, R. (1994). Redefining the goals of protein secondary structure prediction, *Journal of Molecular Biology*, 235:13-26.

Schmidler S, Liu J, Brutlag D. (2000). Bayesian segmentation of protein secondary structure, *Journal of Computational Biology*; 2(1-2):233-48.

Simons K. T., Kooperberg C., Huang E. and Baker D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions, *Journal of Molecular Biology*., 268: 209-25.

KEY TERMS

Bayesian Inference: Bayesian inference is statistical inference in which evidence or observations are used to update or to newly infer the probability that a hypothesis may be true.

Decision Fusion: The process of combining classifiers is called decision fusion. Results from different methods, algorithms, sources or classifiers can often be combined (fused) to give estimates of a better quality than could be obtained from any of the individual sources alone.

Genetic Algorithm: Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

Neural Network: A Neural Network is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.

Profile-Based Neural Network: This type of neural network configuration results when we feed the multiple alignments in the form of a sequence profile (for each position an amino acid frequency vector is fed to the network) instead of a base sequence to a neural network.

Protein: A large molecule composed of one or more chains of amino acids in a specific order determined by the base sequence of nucleotides in the DNA coding for the protein.

Secondary Structure: In biochemistry and structural biology, secondary structure is the general three-dimensional form of local segments of biopolymers such as proteins and nucleic acids (DNA/RNA).

Simulated Annealing Algorithm: Simulated annealing (SA) is a generic probabilistic meta-algorithm

for the global optimization problem, namely locating a good approximation to the global optimum of a given function in a large search space.

Prototype Based Classification in Bioinformatics

Frank-M. Schleif
University of Leipzig, Germany

Thomas Villmann
University of Leipzig, Germany

Barbara Hammer
Technical University of Clausthal, Germany

INTRODUCTION

Bioinformatics has become an important tool to support clinical and biological research and the analysis of functional data, is a common task in bioinformatics (Schleif, 2006). Gene analysis in form of micro array analysis (Schena, 1995) and protein analysis (Twyman, 2004) are the most important fields leading to multiple sub *omics*-disciplines like pharmacogenomics, glyco-proteomics or metabolomics. Measurements of such studies are high dimensional functional data with few samples for specific problems (Pusch, 2005). This leads to new challenges in the data analysis. Spectra of mass spectrometric measurements are such functional data requiring an appropriate analysis (Schleif, 2006). Here we focus on the determination of classification models for such data. In general, the spectra are transformed into a vector space followed by training a classifier (Haykin, 1999). Hereby the functional nature of the data is typically lost. We present a method which takes this specific data aspects into account. A wavelet encoding (Mallat, 1999) is applied onto the spectral data leading to a compact *functional* representation. Subsequently the Supervised Neural Gas classifier (Hammer, 2005) is applied, capable to handle functional metrics as introduced by Lee & Verleysen (Lee, 2005). This allows the classifier to utilize the functional nature of the data in the modelling process. The presented method is applied to clinical proteome data showing good results and can be used as a bioinformatics method for biomarker discovery.

BACKGROUND

Applications of mass spectrometry (ms) in clinical proteomics have gained tremendous visibility in the scientific and clinical community (Villanueva, 2004) (Ketterlinus, 2005). One major objective is the search for potential classification models for cancer studies, with strong requirements for validated signal patterns (Ransohoff, 2005). Primal optimistic results as given in (Petricoin, 2002) are now considered more carefully, because the complexity of the task of biomarker discovery and an appropriate data processing has been observed to be more challenging than expected (Ransohoff, 2005). Consequently the main recent work in this field is focusing on optimization and standardisation. This includes the biochemical part (e.g. Baumann, 2005), the measurement (Orchard, 2003) and the subsequently data analysis (Morris, 2005)(Schleif 2006).

PROTOTYPE BASED ANALYSIS IN CLINICAL PROTEOMICS

Here we focus on classification models. A powerful tool to achieve such models with high generalization abilities is available with the prototype based Supervised Neural Gas algorithm (SNG) (Villmann, 2002). Like all nearest prototype classifier algorithms, SNG heavily relies on the data metric d, usually the standard Euclidean metric. For high-dimensional data as they occur in proteomic patterns, this choice is not adequate due to two reasons: first, the functional nature of the data should be kept as far as possible. Second the noise present in the data set accumulates and likely disrupts the classification when taking a standard Euclidean

approach. A functional representation of the data with respect to the used metric and a weighting or pruning of especially (priory not known) irrelevant function parts of the inputs, would be desirable. We focus on a functional distance measure as recently proposed in (Lee, 2005) referred as functional metric. Additionally a feature selection is applied based on a statistical pre-analysis of the data. Hereby a discriminative data representation is necessary. The extraction of such discriminant features is crucial for spectral data and typically done by a parametric peak picking procedure (Schleif, 2006). This peak picking is often spot of criticism, because peaks may be insufficiently detected and the functional nature of the data is partially lost. To avoid these difficulties we focus on a wavelet encoding. The obtained wavelet coefficients are sufficient to reconstruct the signal, still containing all relevant information of the spectra, but are typically more complex and hence a robust data analysis approach is needed. The paper is structured as follows: first the bioinformatics methods are presented. Subsequently the clinical data are described and the introduced methods are applied in the analysis of the proteome spectra. The introduced method aims on a replacement of the classical three step procedure of denoising, peak picking and feature extraction by means of a compact wavelet encoding which gives a more natural representation of the signal.

BIOINFORMATIC METHODS

The classification of mass spectra involves in general the two steps peak picking to locate and quantify positions of peaks and feature extraction from the obtained peak list. In the first step a number of procedures as baseline correction, denoising, noise estimation and normalization are applied in advance. Upon these prepared spectra the peaks have to be identified by scanning all local maxima. The procedure of baseline correction and recalibration (alignment) of multiple spectra is standard, and has been done here using ClinProTools (Ketterlinus, 2006). As an alternative we propose a feature extraction procedure preserving all (potentially small) peaks containing relevant information by use of the discrete wavelet transformation (DWT). The DWT has been done using the Matlab Wavelet-Toolbox (see http://www.mathworks.com). Due to the local analysis property of wavelet analysis the features can still be related back to original mass position in the spectral

data which is essential for further biomarker analysis. For feature selection the Kolmogorov-Smirnoff test (KS-test) (Sachs, 2003) has been applied. The test was used to identify features which show a significant ($p < 0.01$) discrimination between the two groups (cancer, control). In (Waagen, 2003) also a generalization to a multiclass experiment is given. The now reduced data set has been further processed by SNG to obtain a classification model with a *small* ranked set of features. The whole procedure has been cross-validated in a 10-fold cross validation.

WAVELET TRANSFORMATION IN MASS SPECTROMETRY

Wavelets have been developed as powerful tools (Rieder, 1998) used for noise removal and data compression. The discrete version of the continuous wavelet transform leads to the concept of a multi-resolution analysis (MRA). This allows a fast and stable wavelet analysis and synthesis. The analysis becomes more precise if the wavelet shape is adapted to the signal to be analyzed. For this reason one can apply the so called bi-orthogonal wavelet transform (Cohen, 1992), which uses two pairs of scaling and wavelet functions. One is for the decomposition/analysis and the other one for reconstruction/synthesis, giving a higher degree of freedom for the shape of the scaling and wavelet function. In our analysis such a smooth synthesis pair was chosen. It can be expected that a signal in the time domain can be represented by a small number of a relatively large set of coefficients from the wavelet domain. The spectra are reconstructed in dependence of a certain approximation level L of the MRA. The denoised spectrum looks similar to the reconstruction as depicted in Figure 1.

One obtains approximation- and detail-coefficients (Cohen, 1992). The approximation coefficients describe a generalized peak list, encoding primal spectral information. For linear MALDI-TOF spectra a device resolution of $500-800Da$ can be expected. This implies limits to the minimal peak width in the spectrum and hence, the reconstruction level of the Wavelet-Analysis should be able to model corresponding peaks. A level $L = 4$ is appropriate for our problem (see Figure 1). Applying this procedure including the KS-test on the spectra with an initial number of 22306 measurement points per spectrum one obtains 602 wavelet coefficients

Figure 1. Wavelet reconstruction of the spectra with L = 4, 5, x-mass positions, y-arbitrary unit. Original signal - solid line. One observes for L = 5 (right plot) the peak approximate is to rough.

used as representative features per spectrum, still allowing a reliable functional representation of the data. The coefficients were used to reconstruct the spectra and the final functional representation of the signal.

PROTOTYPE CLASSIFIERS

Supervised Neural Gas (SNG) is considered as a representative for prototype based classification approaches as introduced by Kohonen (Kohonen, 1995). Different prototype classifiers have been proposed so far (Kohonen, 1995) (Sato, 1996) (Hammer, 2005) (Villmann, 2002) as improvements of the original approach. The SNG has been introduced in (Villmann, 2002) and combines ideas from the Neural Gas algorithm (NG) introduced in (Martinetz, 1993) with the Generalized learning vector quantizer (GLVQ) as given in (Sato, 1996).

Subsequently we give some basic notations and remarks to the integration of alternative metrics into Supervised Neural Gas (SNG). Details on SNG including convergence proofs can be found in (Villmann, 2002). Let us first clarify some notations: Let c_v *in L* be the label of input **v**, *L* a set of labels (classes). Let *V in* R^{DV} be a finite set of inputs **v**. LVQ uses a fixed number of prototypes (weight vectors, codebook vectors) for each class. Let $\mathbf{W}=\{\mathbf{w_r}\}$ be the set of all codebook vectors and $c_\mathbf{r}$ be the class label of $\mathbf{w_r}$. Furthermore, let $\mathbf{W}c=\{\mathbf{w_r}/c_\mathbf{r} = c\}$ be the subset of prototypes assigned to class *c in L*. The task of vector quantization is realized by the map Ψ as a winner-take-all rule, i.e. a stimulus vector **v** *in V* is mapped onto that prototype **s** the pointer $\mathbf{w}_s$ of which is closest to the presented stimulus vector **v**, measured by a distance d_λ (**v**,**w**). d_λ (**v**,**w**) is an arbitrary differentiable similarity measure

which may depend on a parameter vector λ. For the moment we take λ as fixed. The neuron **s** (**v**) is called winner or best matching unit. If the class information of the weight vector is used, the above scheme generates decision boundaries for classes (details in (Villmann, 2002)). A training algorithm should adapt the prototypes such that for each class *c in L*, the corresponding codebook vectors **W***c* represent the class as accurately as possible. Detailed equations and cost function for SNG are given in (Villmann, 2002). Here it is sufficient to keep in mind that in the cost function of SNG the distance measure can be replaced by an arbitrary (differentiable) similarity measure, which finally leads to new update formulas for the gradient descent based prototype updates.

Incorporation of a functional metric to SNG As pointed out before, the similarity measure d_λ (**v**,**w**) is only required to be differentiable with respect to λ and **w**. The triangle inequality has not to be fulfilled necessarily (Hammer, 2005). This leads to a great freedom in the choice of suitable measures and allows the usage of non-standard metrics in a natural way. For spectral data, a functional metric would be more appropriate as given in (Lee, 2005). The obtained derivations can be plugged into the SNG equations leading to SNG with a functional metric, whereby the data are functions represented by vectors and, hence, the vector dimensions are spatially correlated. Common vector processing does not take this spatial order of the coordinates into account. As a consequence, the functional aspect of spectral data is lost. For proteome spectra the order of signal features (peaks) is due to the nature of the underlying biological samples and the measurement procedure. The masses of measured chemical compounds are given ascending and peaks encoding chemical structures with a higher mass follow chemical structures with lower masses.

In addition, multiple peaks with different masses may encode parts of the same chemical structure and, hence, are correlated. Lee proposed an appropriate norm with a constant sampling period τ:

$$\mathcal{L}_p^{fc}(\mathbf{v}) = \left(\sum_{k=1}^{D} \left(A_{k-1}(\mathbf{v}) + A_{k+1}(\mathbf{v}) \right)^p \right)^{\frac{1}{p}}$$

with

$$A_k(\mathbf{v}) = \begin{cases} \frac{\tau}{2}|v_k| & \text{if } 0 \le v_k v_{k-1} \\ \frac{\tau}{2}\frac{v_k^2}{|v_k|+|v_{k-1}|} & \text{if } 0 > v_k v_{k-1} \end{cases} \quad B_k(\mathbf{v}) = \begin{cases} \frac{\tau}{2}|v_k| & \text{if } 0 \le v_k v_{k+1} \\ \frac{\tau}{2}\frac{v_k^2}{|v_k|+|v_{k+1}|} & \text{if } 0 > v_k v_{k+1} \end{cases}$$

are respectively of the triangles on the left and right sides of x_i. Just as for Lp, the value of p is assumed to be a positive integer. At the left and right ends of the sequence, x_0 and x_D are assumed to be equal to zero. The derivatives for the functional metric taking $p = 2$ are given in (Lee, 2005). Now we consider the scaled functional norm where each dimension $(0, 1]$, v_i is scaled by a parameter $\lambda_i > 0$ and all λ_i sum up to 1:

$$\mathcal{L}_p^{fc}(\lambda\mathbf{v}) = \left(\sum_{k=1}^{D} \left(A_{k-1}(\lambda\mathbf{v}) + A_{k+1}(\lambda\mathbf{v}) \right)^p \right)^{\frac{1}{p}} \quad (9)$$

with

$$A_k(\lambda\mathbf{v}) = \begin{cases} \frac{\tau}{2}\lambda_k|v_k| & \text{if } 0 \le v_k v_{k-1} \\ \frac{\tau}{2}\frac{\lambda_k^2 v_k^2}{\lambda_k|v_k|+\lambda_{k-1}|v_{k-1}|} & \text{else} \end{cases} \quad B_k(\lambda\mathbf{v}) = \begin{cases} \frac{\tau}{2}\lambda_k|v_k| & \text{if } 0 \le v_k v_{k+1} \\ \frac{\tau}{2}\frac{\lambda_k^2 v_k^2}{\lambda_k|v_k|+\lambda_{k+1}|v_{k+1}|} & \text{else} \end{cases} \quad (10)$$

The prototype update changes to:

$$\frac{\partial \delta_2^2(\mathbf{x}, \mathbf{y}, \lambda)}{\partial x_k} = \frac{\tau^2}{2}(2 - U_{k-1} - U_{k+1})(V_{k-1} + V_{k+1})\triangle_k$$

with

$$U_{k-1} = \begin{cases} 0 \text{ if } 0 \le \triangle_k \triangle_{k-1} \\ \left(\frac{\lambda_{k-1}\triangle_{k-1}}{\lambda_k|\triangle_k|+\lambda_{k-1}|\triangle_{k-1}|} \right)^2 \text{ else} \end{cases}, \quad U_{k+1} = \begin{cases} 0 \text{ if } 0 \le \triangle_k \triangle_{k+1} \\ \left(\frac{\lambda_{k+1}\triangle_{k+1}}{\lambda_k|\triangle_k|+\lambda_{k+1}|\triangle_{k+1}|} \right)^2 \text{ else} \end{cases}$$

$$V_{k-1} = \begin{cases} 1\lambda_k \text{ if } 0 \le \triangle_k \triangle_{k-1} \\ \frac{\lambda_k|\triangle_k|}{\lambda_k|\triangle_k|+\lambda_{k-1}|\triangle_{k-1}|} \text{ else} \end{cases}, \quad V_{k+1} = \begin{cases} 1\lambda_k \text{ if } 0 \le \triangle_k \triangle_{k+1} \\ \frac{\lambda_k|\triangle_k|}{\lambda_k|\triangle_k|+\lambda_{k+1}|\triangle_{k+1}|} \text{ else} \end{cases}$$

And $\triangle_k = x_k - y_k$ using this parameterization one can emphasize/neglect different parts of the function for classification.

ANALYSIS OF PROTEOMIC DATA

The proposed data processing scheme is applied to clinical ms spectra taken from a cancer study (45 cancer, 50 control samples). Sample preparation and profile spectra analysis were carried out using the CLINPROT system (Bruker Daltonik, Bremen, Germany [BDAL]). The preprocessed set of spectra and the corresponding wavelet coefficients are then analyzed using the SNG

extended by a functional metric. We reconstructed the spectra based upon the discriminative wavelet coefficients determined by the Kolmogorov-Smirnoff test as explained above and used corresponding intensities as features. We used all features for the parameterized functional norm i.e. all $\lambda_i = 1$. The original signal with approx. 22000 sampling points had been processed with only 600 remaining points still encoding the significant parts of the signal relevant for discrimination between the classes. The SNG classifier with functional metric obtains a crossvalidation accuracy of 84% using functional metric and 82% by use of standard Euclidean metric. The results from the wavelet processed spectra are slightly better than using standard peak lists, with 81% crossvalidation accuracy.

FUTURE TRENDS

The proposed method generates a compact but still complex functional representation of the spectral data. While the bior3.7 wavelet gives promising results they are still not optimal, due to signal oscillations, leading to negative intensities in the reconstruction. Further, the functional nature of the data motivates the usage of a functional data representation and similarity calculation but there are also spectra regions encoded which do not contain meaningful biological information but measurement artefacts. In principle it should be possible to remove this overlaying artificial function from the real signal. Further it could be interesting to incorporate additional knowledge about the peak width, which is increasing over the mass axis.

CONCLUSION

The presented interpretation of proteome data demonstrate that the functional analysis and model generation using SNG with functional metric in combination with a wavelet based data pre-processing provides an easy and efficient detection of classification models. The usage of wavelet encoded spectra features is especially helpful in detection of small differences which maybe easily ignored by standard approaches as well as to generate a significant reduced number of points needed in further processing steps. The signal must not be shrinked to peak lists but could be preserved in its functional representation. SNG was able to process

high-dimensional functional data and shows good regularization. By use of the Kolmogorov-Smirnoff test we found a ranking of the features related to mass positions in the original spectrum which allows for identification of most relevant feature dimensions and to prune irrelevant regions of the spectrum. Alternatively one could optimize the scaling parameters of the functional norm directly during classification learning by so called relevance learning as shown in (Hammer, 2005) for scaled Euclidean metric. Conclusively, wavelet spectra encoding combined with SNG and a functional metric is an interesting alternative to standard approaches. It combines efficient model generation with automated data pre-treatment and intuitive analysis.

REFERENCES

Baumann, S., Ceglarek, U., Fiedler, G.M. & Lembcke, J. (2005) Standardized approach to proteomic profiling of human serum based magnetic bead separation and matrix-assisted laser esorption/ionization time-of flight mass spectrometry. Clinical Chemistry, 51, 973—980

Cohen, A., Daubechies, I. & Feauveau, J.-C. (1992) Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45(5):485–560.

Hammer, B., Strickert, M. & Villmann, T. (2005) Supervised neural gas with general similarity measure. *Neural Processing Letters*, 21(1):21–44.

Haykin, S. (1999). *Neural Networks* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Ketterlinus, R., Hsieh, S-Y., Teng, S-H., Lee, H. & Pusch, W. (2005) Fishing for biomarkers: analyzing mass spectrometry data with the new clinprotools software. *Biotechniques*, 38(6):37–40, 2005.

Kohonen, T. (1995). *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, (2nd Ext. Ed. 1997).

Lee, J. & Verleysen, M. (2005) Generalizations of the lp norm for time series and its application to self-organizing maps. In Marie Cottrell, editor, *5th Workshop on Self-Organizing Maps*, volume 1, pages 733–740.

Mallat, S (1998) A wavelet tour of signal processing. San Diego, CA: Academic Press.

Martinetz, T., Berkovich, S. & Schulten, K. (1993) 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569.

Morris, J., Coombes, K., Koomen, J., Baggerly, K. & Kobayashi, R. (2005) Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. Bioinformatics 21(9), 1764—1775.

Orchard, S. Hermjakob, H. & Apweiler, R. (2003) The Proteomcs Standards Initiative, Proteomics, 3, 1274--1376.

Pusch, W., Flocco, M., Leung, S.M., Thiele, H. Kostrzewa, M.(2003). Mass spectrometry-based clinical proteomics. Pharmacogenomic, 4, 463--476.

Petricoin, E.F., Ardekani, A., Hitt, B. Levine, P. (2002) Use of proteomic patterns in serum to identify ovarian cancer. Lancet, 359, 572—577.

Ransohoff, D. F. (2005) Lessons from controversy: ovarian cancer screening and serum proteomics, J Natl Cancer Inst, 97, 315—319, 2005.

Rieder, A. Louis, A.K. & Maaß, P. (1998) *Wavelets: Theory and Applications*. Wiley.

Sachs, L. (2003) Angewandte Statistik. Springer.

Sato, A. & Yamada, K. (1996) Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA.

Schena, M., Shalon, D., Davis, R.W. & Brown, P.O. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray, Science, 270, 5235: 467—470.

Schleif, F.-M.(2006) Prototype based Machine Learning for Clinical Proteomics. Technical University Clausthal, PhD-Thesis.

Twyman, R.M. Principles of proteomics BIOS Scientific Publishers, NY,2004.

Villanueva, J., Philip, J., Entenberg, D. & Chaparro, C.A. (2004) Serum peptide profiling by magnetic particle-assisted, automated sample processing and maldi-tof mass spectrometry. *Anal. Chem.*, 76:1560–1570.

Villmann, T. & Hammer, B. (2002) Supervised neural gas for learning vector quantization. In D. Polani, J. Kim, and T. Martinetz, editors, *Proc. of the 5th GermanWorkshop on Artificial Life (GWAL-5)*, pages 9–16. Akademische Verlagsgesellschaft - infix - IOS Press, Berlin.

Waagen, D.E., Cassabaum, M.L. , Scott, C. & Schmitt, H.A. (2003) Exploring alternative wavelet base selection techniques with application to high resolution radar classification. In *Proc. of the 6th Int. Conf. on Inf. Fusion (ISIF'03)*, pages 1078–1085. IEEE Press.

KEY TERMS

Bioinformatics: Generic term of a research field as well as a set of methods used in computational biology or medicine to analyse multiple kinds of biological or clinical data. It combines the disciplines of computer science, artificial intelligence, applied mathematics, statistics, biology, chemistry and engineering in the field of biology and medicine. Typical research subjects are problem adequate data pre-processing of measured biological sample information (e.g. data cleaning, alignments, feature extraction), supervised and unsupervised data analysis (e.g. classification models, visualization, clustering, biomarker discovery) and multiple kinds of modelling (e.g. protein structure prediction, analysis of expression of gene, proteins, gene/protein regulation networks/interactions) for one or multidimensional data including time series. Thereby the most common problem is the high dimensionality of the data and the small number of samples which in general make standard approach (e.g. classical statistic) inapplicable.

Biomarker: Mainly in clinical research one goal of experiments is to determine patterns which are predictive for the presents or prognosis of a disease state, frequently called biomarker. Biomarkers can be single or complex (pattern) indicator variables taken from multiple measurements of a sample. The ideal biomarker has a high sensitivity, specificity and is reproducible (under standardized conditions) with respect to control experiments in other labs. Further it can be expected that the marker is vanishing or changing during a treatment of the disease.

Clinical Proteomics: Proteomics is the field of research related to the analysis of the proteome of an organism. Thereby, clinical proteomics is focused on research mainly related to disease prediction and prognosis in the clinical domain by means of proteome analysis. Standard methods for proteome analysis are available by Mass spectrometry.

Mass Spectrometry: An analytical technique used to measure the mass-to-charge ratio of ions. In clinical proteomics mass spectrometry can be applied to extract fingerprints of samples (like blood, urine, bacterial extracts) whereby semi-quantitative intensity differences between sample cohorts may indicate biomarker candidates

Prototype Classifiers: Are a specific kind of neural networks and related to the kNN classifier. The classification model consists of so called prototypes which are representatives for a larger set of data points. The classification is done by a nearest neighbour classification using the prototypes. Nowadays prototype classifiers can be found in multiple fields (robotics, character recognition, signal processing or medical diagnosis) trained to find (non)linear relationships in data.

Relevance Learning: A method, typically used in supervised classification, to determine problem specific metric parameter. With respect to the used metric and learning schema univariate, correlative and multivariate relations between data dimensions can be analyzed. Relevance learning typically leads to significantly improved, problem adapted metric parameters and classification models.

Wavelet Analysis: Method used in signal processing to analyse a signal by means of frequency and local information. Thereby the signal is encoded in a representation of wavelets, which are specific kinds of mathematical functions. The Wavelet encoding allows the representation of the signal at different resolutions, the coefficients contain frequency information but can also be localized in the signal.

Randomized Hough Transform

Lei Xu
Chinese University of Hong Kong & Peking University, China

Erkki Oja
Helsinki University of Technology, Finland

INTRODUCTION

Proposed in 1962, the Hough transform (HT) has been widely applied and investigated for detecting curves, shapes, and motions in the fields of image processing and computer vision. However, the HT has several shortcomings, including high computational cost, low detection accuracy, vulnerability to noise, and possibility of missing objects. Many efforts target at solving some of the problems for decades, while the key idea remains more or less the same. Proposed in 1989 and further developed thereafter, the Randomized Hough Transform (RHT) manages to considerably overcome these shortcomings via innovations on the fundamental mechanisms, with random sampling in place of pixel scanning, converging mapping in place

of diverging mapping, and dynamic storage in place of accumulation array. This article will provides an overview on advances and applications of RHT in the past one and half decades.

BACKGROUND

Taking straight line detection as an example, the upper part of Fig.1 shows the key idea of the Hough Transform (HT) (Hough, 1962) . A set of points on a line $y=kx+b$ in the image are mapped into a set of lines across a point (k, b) in the parameter space. A uniform grid is located on a window in the (k, b) space, with an accumulator $a(k, b)$ at each bin. As each point (x,y) on the image is mapped into a line in the (k, b) space, every associated accumulator $a(k, b)$ is incremented by 1. We can detect

Figure 1. From hough transform to randomized hough transform
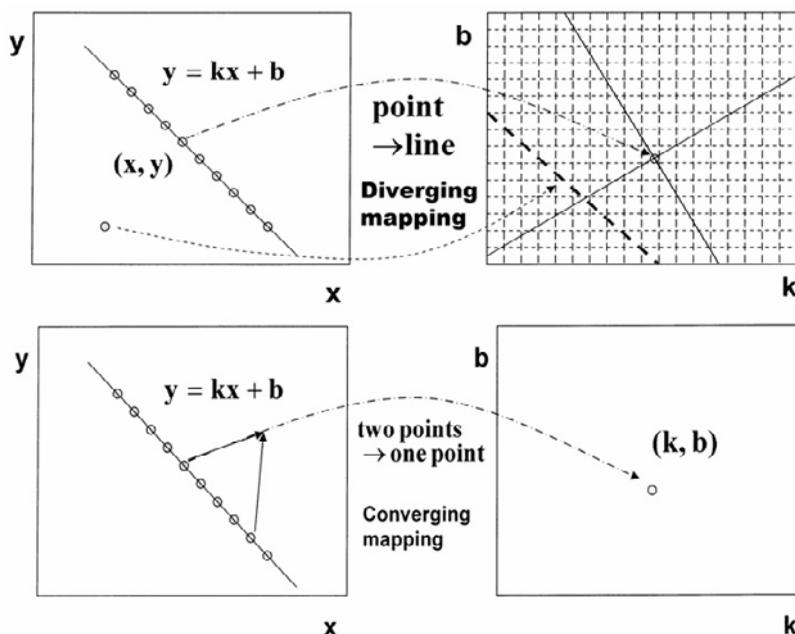
lines by finding every accumulator with it's score $a(k, b)$ larger than a given threshold.

The Hough Transform was brought to the attention of the mainstream image processing community by Rosenfeld (1969). Then Duda and Hart (1972) not only introduced the polar parameterization technique for more efficient line detection, but also demonstrated how a circle can be detected. Kimme, Ballard and Sklansky (1975) made circular curve detection significantly more effective by using the gradient information of pixels. Merlin and Faber (1975) showed how the HT could be generalized to detect an arbitrary shape at a given orientation and a given scale. Ballard (1981) eventually generalized the HT to detect curves of a given arbitrary shape for any orientation and any scale. Since then, a lot of applications, variants and extensions of the HT have been published in the literature. A survey on these developments of the HT is given by Illingworth and Kittler (1988).

However, the HT has several critical drawbacks as follows:

a. All pixels are mapped, and every bin in the grid needs an accumulator. If there are d parameters, each represented by M bins or grid points, one needs M^d accumulators.

b. To reduce the computational cost, quantization resolution cannot be high, which blurs the peaks and leads to low detection accuracy.

c. Each pixel activates every accumulator located on a line, but there is only one that represents the correct one while all the others are disturbances.

d. If the grid window is set inappropriately, some objects may locate outside the window and thus cannot be detected.

e. Disturbing and noisy pixels cause many interfering accumulations.

Many efforts have been made to alleviate these problems. Using the gradient information of pixels is one of them. Another is analyzing noise and error sensitivity (vanVeen, 1981; Brown, 1983; Grimson & Huttenlocher, 1990). The third is the use of hierarchical voting accumulation (Li, Lavin & LeMaster, 1986) or multiresolution (Atiquzzaman, 1992). Yet another is improving the effect of quantization through the use of kernels (Palmer, Petrou, & Kittler, 1993) or error propagation analysis (Ji & Haralick, 2001), as well as hypothesis testing (Princen, Illingworth, & Kittler,

1994). However, none of these suggestions offer any fundamental changes to the key mechanisms of HT.

Proposed in 1989 and further investigated thereafter (Xu, Oja, & Kultanen, 1990; Xu & Oja, 1993), the Randomized Hough Transform (RHT) tackles the above problems by using a fundamental innovation: the one-to-many diverging mapping from the image space to the parameter (accumulator) space, as shown in the upper part of Fig.1(a), is replaced by a many-to-one converging mapping, as shown in the bottom part of Fig.1(a). This fundamental change further enables several joint improvements, such as a random sampling in place of pixel scanning, a small size dynamic storage in place of the array of M^d accumulators, and an adaptive detection in place of enumerating all the pixels and picking those accumulators with scores larger than a threshold. As a result, not only time and storage complexity have been reduced significantly, but also the detection accuracy has been improved considerably.

Subsequently, many studies have been made on RHT. On one hand, there are various real applications such as medical images (Behrens, Rohr, & Siegfried, 2003), range images (Ding, et al, 2005), motion detection (Heikkonen, 1995), object tracking for a mobile robot (Jean & Wu, 2004), soccer robot (Claudia, Rous, & Kraiss, 2004), mine detection (Milisavljevic, 1999), and others (Chutatape & Guo, 1999). On the other hand, there are also many further developments on RHT, including an efficient parameterization for ellipse detection (McLaughlin, 1998), extension to motion detections (Kalviainen, Oja, & Xu, 1991; Xu, 2007), the uses of local gradient information, local connectivity and neighbor-orientation for further improvements (Brailovsky, 1999; Kalviainen & Hirvonen, 1997), an integration with error propagtion analysis (Ji & Xie, 2003), a modification of random sampling to importance sampling (Walsha & Raftery, 2002), and others (Xu, 2007). Due to space limit, it is not possible to provide a complete survey here. An early review on RHT variants is referred to (Kalviainen, Hirvonen, Xu, & Oja, 1995), and recent elaborations on RHT are referred to (Xu, 2007).

It may also need to be mentioned that the literature on RHT studies often includes studies under the name of probabilistic HT (Bergen & Shvaytser, 1991; Kiryati, Eldar & Bruckstein, 1991) that also suggests to use a random sampling to replace the scanning in the implementation of the standard HT and thus shares one

of the previously mentioned RHT features. However, it will not lose too much generality to regard it as a degenerated case of RHT for an understanding purpose, though there are some detailed differences.

BASIC RHT MECHANISMS AND CHARACTERISTICS

As shown in Fig.1, one pixel is mapped into all the points on a line passing (k,b) by the diverging mapping mechanism of HT, which actually incurs the above drawbacks (a)-(e). RHT replaces this mechanism with a converging mapping mechanism such that two or more pixels are picked to jointly determine a line, i.e., mapped into one point (k,b). By this mechanism, different points on the same line $y=kx+b$ will hit the same point (k,b), without creating a great number of false accumulations. Also, the feature of being mapped

into one point at a time makes it possible to construct accumulators dynamically, with no need of laying a grid on a pre-specified window. We only need to accumulate $a(k, b)$ at those locations activated by the converging mappings. Also, quantization resolution may vary for different locations, and each quantization bin can be replaced by a kernel. As a result, the drawbacks (b),(c),(d) no longer exist.

Without considering the quantization effect, if there is a line consisting of n pixels on an image, we get a peak with n counts in its accumulated scores. Assume that in its neighbour there is another peak of false line consisting of $m < n$ pixels, then the ratio n/m describes a signal/noise ratio of a reliable detection by HT. In RHT, assuming that we exhaust all the possible pairs of pixels, the voting counts for the line will be $n(n-1)/2$ while the voting counts for the disturbing false line will be $m(m-1)/2$, i.e., the signal/noise ratio becomes $\frac{n(n-1)}{m(m-1)}$ that is $\frac{n-1}{m-1}$ times increased compared

Table 1. Missing probability versus false alarm probability

THE DETECTING RULE	
On an image that consists of N pixels Detect a point $\theta \in \Theta$ as a line if it is hit by more than k_0 times.	
MISSING PROBABILITY	**FALSE-ALARM PROBABILITY**
Consider a line consisting of n pixels, a trial of randomly sampling two pixels has a probability $p_c = \frac{n(n-1)}{N(N-1)}$ that both pixels come from the line, i.e., the line is successfully hit in the parameter space by a probability p_c. After M trials, the number of being successfully hit is a variable ξ in a binomial distribution, $p_c(\xi=k)=C_M^k p_c^k (1-p_c)^{M-k}$ see eqn.(4b) in (Xu & Oja, 1993). A risk of missing this line by the detecting rule has a probability $P_{miss} = \sum_{k=0}^{k_0} p_c(\xi=k)$ controlling it below a pre-specified rate, we can determine a lower bound $M > M_c$.	Consider a false line consisting of m pixels, a trial of randomly sampling two pixels has a probability $p_r = \frac{m(m-1)}{N(N-1)}$ that both pixels come from the line, i.e., the line is successfully hit in the parameter space by a probability p_r. After M trials, the number that it is hit is a variable ξ in a binomial distribution too, $p_r(\xi=k)=C_M^k p_r^k (1-p_r)^{M-k}$ In this case, taking a point $\theta \in \Theta$ that is hit by more than k_0 times as a line has a risk of taking a false line as a solution with a probability $P_{false} = 1 - \sum_{k=0}^{k_0} p_r(\xi=k)$ controlling it below a pre-specified rate, we can determine an upper bound $M < M_r$.

as discussed in (Xu, 2007), a solution can also be obtained by either a least square fitting, an L_p norm fitting, or by maximum likelihood estimation. Sometimes, it may even consider under-constrained equations by taking less samples, from which a parametric curve or surface in Θ is obtained to implement an array based accumulation similar to HT.

Second, there are also alternatives for Step 2 and Step 3. One extreme is returning to an array based accumulation. The other extreme is that all the mapped points in Θ are stored as they are, and either cluster analysis or kernel based density estimation is made on them to find cluster centres and density degrees for detecting curves or objects. Between the two extremes, we may consider a trade off or their combination (Xu, 2007). Third, Step 4 can also be performed with different choices, including a δ-band test, a fitting error threshold, and a hypothesis testing (Xu, 2007).

Moreover, instead of checking candidate solution every time t, we can let the procedure run until $t = M_c$, put those accumulators with $a(\theta) > k_0$ into S_θ as candidate solutions and examine these candidates at Step 4. Also, checking and examining candidates can be made per a pre-specified period. Furthermore, gradient information in a grey image may also improve the converging mapping.

The last but not the least, RHT has also be extended to detect objects by a template as shown in Fig.2.

FUTURE TRENDS

Challenges to RHT mainly come from the effects of noise and quantization. Two types of noise are shown in Fig.3. The first type is in Fig.3(a) with disturbing pixels added but the original pixels unaffected. This noise type may reduce the signal/noise ratio, resulting in more computing time and space. However, the accuracy of the detected line will be not affected. The second type is in Fig.3(b), with some original pixels deviated from the exact line. The quantization effect can be regarded as a special case of this type that uniformly distributed noise is added to the coordinates of pixels. The second type not only reduces the signal/noise ratio but also makes the detected line inaccurate. As yet, there lacks a systematic theoretical analysis on how the solution accuracy will be affected by this second type. More importantly, theoretical guides are lacking on how to control the accuracy of detected curves and objects.

The tasks of detecting curves and objects can also be performed from the perspective of mixture based learning, which is much more robust in the case of the second type of noise (Xu, 2003; Liu, Qiao, & Xu, 2006; Xu, 2007). Solving pattern recognition tasks by machine learning approaches is a popular trend in the past decade and currently. Actually, the machine learning perspective are complementary to the perspective

Figure 2. Use a template to match a shape via translation μ, rotation ϕ and scaling λ

each pixel $u(x, y)$ on the image relates to a corresponding point v on the template via
$$u = \lambda R(\phi)(v - \mu),$$
$R(\phi)$ is a rotation matrix of angle ϕ.

CONVERGING MAPPING :
randomly sampling $m \geq 2$ pairs of (u, v) with u from the image and v from the template, the parameters λ, ϕ, μ can be solved jointly by $m \geq 2$ matrix equations for each pair.

Alternatively, setting $\lambda = 1$, the motion of an object can be detected by the displacement μ and the rotation ϕ after a time interval.

Figure 3. Different effects by two types of noises

(a)　　　　　　　　(b)
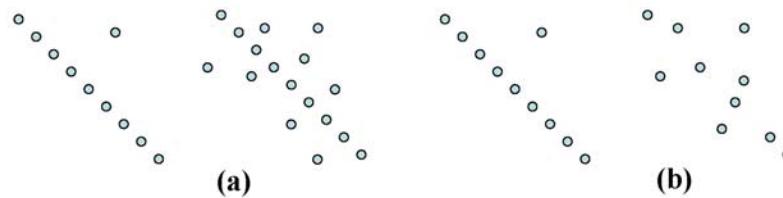
of HT/RHT type evidence accumulation. A trend is integrating the strengths of both.

CONCLUSION

This article provides not only a brief overview on nearly two decade developments and applications of RHT for detecting curves, shapes, and motions, but also a tutorial and re-elaboration on basic mechanisms, variants, and extensions of RHT, as well as challenges and future trends of RHT studies. Recently, a general problem solving paradigm has been developed and implemented by an integration of five essential mechanisms (Xu, 2007). Not only the difference between the machine learning perspective and HT/RHT perspective can be understood via handling two coupled core tasks, namely amalgamating evidences and discriminating differences, but also different implementations of these mechanisms and differences in a specific integration may bring us new results and potential directions for future studies.

ACKNOWLEDGMENT

REFERENCES

Atiquzzaman, M.,(1992), Multiresolution Hough transform-an efficient method of detecting patterns inimages, IEEE Transactions Pattern Analysis Machine Intelligence 14,1090–1095.

Ballard, D.H., (1981), Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition, 13(2),111-122.

Behrens, T., Rohr, K., & Stiehl1, S., H., (2003), Robust Segmentation of Tubular Structures in 3D Medical Images by Parametric Object Detection and Tracking, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 33(4),554-561.

Bergen, J.R., & Shvaytser, H., (1991), A probabilistic algorithm for computing Hough transforms, Journal of Algorithms 12,639–656.

Brailovsky, V., (1999), Fast and robust techniques for detecting straight line segments using local models, Pattern Recognition Letters 20,865-877.

Brown, C.M., (1983), Inherent bias and noise in the Hough transform, IEEE Transactions Pattern Analysis Machine Intelligence 5,493–505.

Chutatape, O., & Guo,L.,(1999), A modified Hough transform for line detection and its performance, Pattern Recognition 32,181–192.

Claudia, G., Rous, M., & Kraiss, K.F., (2004), Real Time Adaptive Colour Segmentation for the RoboCup Middle Size, RoboCup2004, LNAI3276, Springer, 402-410.

Ding, Y.H., et al, (2005), Range image segmentation based on randomized Hough transform, Pattern Recognition Letters 26,2033–2041.

Duda, R.O., & Hart, P.E., (1972), Use of the Hough transform to detect lines and curves in pictures, Communications of the ACM 15(1), 11-15.

Grimson, W.E.L.& Huttenlocher, D.P., (1990), On the sensitivity of the Hough transform for object recogni-

tion, IEEE Transactions Pattern Analysis Machine Intelligence 12,255-274.

Heikkonen, J., (1995), Recovering 3D motion parameters from optical flow field using randomized Hough transform, Pattern Recognition Letters 15,971–978.

Hough, P.V.C., (1962), Method and means for recognizing complex patterns, U.S. Patent 3069654, Dec.18, 1962.

Illingworth, J. & Kittler, J., (1988), A survey of the Hough Transform, Computer Vision Graphics and Image Processing 43, 221-238.

Illingworth, J. & Kittler, J., (1987), The adaptive Hough Transform, IEEE Transactions Pattern Analysis Machine Intelligence 9,690-698.

Kimme,C.D., Ballard,D.H.,& Sklansky, J.,(1975), Finding circles by an array of accumulators, Communications of the ACM 18(2),120-122.

Jean, J.H.. & Wu, T., (2004), Robust visual servo control of a mobile robot for object tracking in shape parameter space, 43rd IEEE Decision & Control Conference, 4016-4021.

Ji, Q. & Xie, Y., (2003), Randomised Hough transform with error propagation for line and circle detection, Pattern Analysis and Application 6,55–64.

Ji, Q. & Haralick, R.Q., (2001), Error propagtion for Hought Transform, Pattern Recognition Letters 22,813-823.

Kalviainen, H. & Hirvonen, P., (1997), An extension to the randomized Hough transform exploiting connectivity, Pattern Recognition Letters, 18(1),77-85.

Kalviainen, H., Hirvonen, P., Xu, L. & Oja, E., (1995), Probabilistic and nonprobabilistic Hough transforms: Overview and comparison, Image Vision Computing 13,239–252.

Kalviainen, H., Oja, E., & Xu, L., (1991), Motion Detection Using Randomized Hough Transform, Proceedings 7th Scandinavian Conference on Image Analysis, 72-79.

Kiryati, N., Eldar, Y., & Bruckstein, A.M., (1991), A probabilistic Hough transform, Pattern Recognition, 24(4): 303–316.

Li, Z, Lavin, M.A., LeMaster, R.J., (1986), Fast Hough transform: a hier-archical approach, Computer Vision, Graph Image Processing 36,139–161.

Liu, Z.Y., Qiao, H. , & Xu, L. , (2006), Multisets Mixture learning based Ellipse Detection, Pattern Recognition, 39,731-735.

McLaughlin, R.A., (1998), Randomized Hough transform: improved ellipse detection with comparison, Pattern Recognition Letters 19(3-4), 299–305.

Milisavljevic, N., (1999), Comparison of three methods for shape recognition in the case of mine detection, Pattern Recognition Letters 20(11–13), 1079–1083.

Olson, C.F., (1999), Constrained Hough transforms for curve detection, Computer Vision and Image Understanding, 73(3),329–345.

Merlin, P.M.& Farber, D.J.,(1975), A parallel mechanism for detecting curves in pictures, IEEE Transactions Computer 24, 96-98.

Palmer, P.L., Petrou, M., & Kittler,J., (1993), A Hough transform algorithm with a 2D hypothesis testing kernel, Computer Vision, Graphics, and Image Processing: Image Understanding 58(2),221–234.

Princen, J., Illingworth, J., & Kittler, J., (1994), Hypothesis testing: A framework for analyzing and optimizing Hough transform performance, IEEE Transactions Pattern Analysis Machine Intelligence 16(4),329–341.

Risse, T., (1989), Hough Transformation for line recognition: complexity of evidence accumulation and cluster detection, Computer Vision Graphics and Image Processing 46, 327-345.

Rosenfeld, A., (1969), *Picture Processing by Computer*, Academic Press, New York.

Shapiro, S.D., & Iannino, A., (1979), Geometric constructions for predicting Hough transform performance, IEEE Transactions Pattern Analysis Machine Intelligence 1(3),310–317.

Walsha, D, & Raftery, A.E., (2002), Accurate and efficient curve detection in images: the importance sampling Hough transform, Pattern Recognition 35,1421–1431.

vanVeen, T.M., & Groen, F.C.A. (1981), Discretization errors in the Hough transform, Pattern Recognition 14(1–6):137–145.

Xu, L (2007), A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving, Pattern Recognition 40, 2129–2153.

Xu, L (2003), Data smoothing regularization, multi-sets-learning, and problem solving strategies, Neural Networks 16, 817-825.

Xu, L., & Oja, E., (1993), Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms and Complexities, Computer Vision, Graphics, and Image Processing: Image Understanding 57, 131-154.

Xu, L. Oja, E., & .Kultanen, P., (1990), A New Curve Detection Method Randomized Hough transform (RHT), Pattern Recognition Letters 11,331-338.

KEY TERMS

δ Band Test: A pixel is said to fall in the δ band of ρ (it denotes a curve or surface) in the image space if the shortest distance from this pixel to ρ is less than a pre-specified threshold δ. Pixels falling in the δ band of ρ are regarded as belonging to ρ, and a δ band test can be designed according to these pixels.

Cluster Analysis: Beyond using an accumulation array, in the cases of a converging mapping, every mapped point in R^κ is memorized. After an enough number of converging mappings, we get a set of points on which cluster analyses can be made to find clusters' centre (mean or median).

Diverging Mapping vs. Converging Mapping: Given pixels of a number m, a set of under-constrained equations specify a curve or manifold of a dimension $\geq \kappa - m$ in R^κ if $m < \kappa$. E.g., from a line $y=kx+b$ passing a given pixel in the image, we have a line b=y-kx in R^2. This case is called diverging mapping because m pixels are mapped diversely to the R^κ space. On the other hand, if $m \geq \kappa$, a unique point in the R^κ space maybe determined by solving a set of joint equations or optimizing a cost when the joint equations are over-constrained, i.e., we have a converging mapping that maps m pixels into one point in R^κ.

Kernel Estimator: Every mapped point is memorized as the centre of a kernel function, e.g., a bell-shaped such as a Gaussian. Collectively, mapped points forms a density estimation for a multi-mode distribution, with each mode in place of the above cluster centre.

Random Sampling: Given a set of N pixels, we take a number m of pixels with each picked randomly with a probability $1/N$. Repeating this sampling by an enough number of times, a global configuration of N pixels will emerge, without enumerating all the N pixels.

Threshold Based Voting vs. Local Maxima Finding: Given a pre-specified threshold, an accumulator in an array is picked if it receives votes larger than the threshold, without considering any neighborhood. Finding a local maximum means to find an accumulator with its votes larger than those of accumulators located in its neighborhood area.

Under-Constrained vs. Over-Constrained Equations: For a parametric equation of κ free parameters, we have a set of under-constrained equations with pixels of a number $m < \kappa$ and a set of over-constrained equations with pixels of a number $m \geq \kappa$ in a non-degenerate way.

Ranking Functions

R

Franz Huber
California Institute of Technology, USA

INTRODUCTION

Ranking functions have been introduced under the name of ordinal conditional functions in Spohn (1988; 1990). They are representations of epistemic states and their dynamics. The most comprehensive and up to date presentation is Spohn (manuscript).

BACKGROUND

The literature on knowledge, belief, and uncertainty in artificial intelligence is divided into two broad classes. In epistemic logic (Hintikka 1961, Halpern & Fagin & Moses & Vardi 1995), belief revision theory (Alchourrón & Gärdenfors & Makinson 1985, Gärdenfors 1988, Rott 2001), and nonmonotonic reasoning (Kraus & Lehmann & Magidor 1990, Makinson 2005) qualitative approaches are used to represent the epistemic state of an agent. In probability theory (Pearl 1988, Jeffrey 2004) and alternatives (Dempster 1968, Shafer 1976, Dubois & Prade 1988) epistemic states are represented quantitatively as degrees of belief rather than yes-or-no beliefs (see Halpern 2003 for an overview). One of the distinctive features of ranking functions is that they are quantitative, but nevertheless induce a notion of yes-or-no belief that satisfies the standard requirements of rationality, viz. consistency and deductive closure.

RANKING FUNCTIONS

Let W be a non-empty set of possibilities or worlds, and let $\mathbf{A}$ be a field of propositions over W. That is, $\mathbf{A}$ is a set of subsets of W that includes the empty set $\varnothing$ ($\varnothing \in \mathbf{A}$) and is closed under complementation with respect to W (if $A \in \mathbf{A}$, then $W \backslash A \in \mathbf{A}$) and finite intersection (if $A \in \mathbf{A}$ and $B \in \mathbf{A}$, then $A \cap B \in \mathbf{A}$). A function ρ from the field $\mathbf{A}$ over W into the natural numbers N extended by ∞, $\rho \colon \mathbf{A} \to N \cup \{\infty\}$, is a (*finitely minimitive*) *ranking function* on $\mathbf{A}$ if and only if for all propositions A, B in $\mathbf{A}$:

1. $\rho(W) = 0$
2. $\rho(\varnothing) = \infty$
3. $\rho(A \cup B) = \min\{\rho(A), \rho(B)\}$

If the field of propositions $\mathbf{A}$ is closed under countable intersection (if $A_1 \in \mathbf{A}, \ldots, A_n \in \mathbf{A}, \ldots, n \in N$, then $A_1 \cap \ldots \cap A_n \cap \ldots \in \mathbf{A}$) so that $\mathbf{A}$ is a σ-field, a ranking function ρ on $\mathbf{A}$ is *countably minimitive* if and only if it holds for all propositions $A_1 \in \mathbf{A}, \ldots A_n \in \mathbf{A}, \ldots$

4. $\rho(A_1 \cup \ldots \cup A_n \cup \ldots) = \min\{\rho(A_1), \ldots, \rho(A_n), \ldots\}$

If the field of propositions $\mathbf{A}$ is closed under arbitrary intersection (if $\mathbf{B} \subseteq \mathbf{A}$, then $\cap \mathbf{B} \in \mathbf{A}$) so that $\mathbf{A}$ is a γ-field, a ranking function ρ on $\mathbf{A}$ is *completely minimitive* if and only if it holds for all sets of propositions $\mathbf{B} \subseteq \mathbf{A}$:

5. $\rho(\cup \mathbf{B}) = \min\{\rho(A) \colon A \in \mathbf{B}\}$

A ranking function ρ on $\mathbf{A}$ is *regular* just in case $\rho(A) < \infty$ for each non-empty or consistent proposition A in $\mathbf{A}$.

The conditional ranking function $\rho(\cdot|\cdot) \colon \mathbf{A} \times \mathbf{A} \to N \cup \{\infty\}$ based on the ranking function ρ on $\mathbf{A}$ is defined such that for all propositions A, B in $\mathbf{A}$:

6. $\rho(A|B) = \rho(A \cap B) - \rho(B)$ if $A \neq \varnothing$, and $\rho(\varnothing|B) = \infty$

$\rho(\cdot|B)$ is a ranking function on $\mathbf{A}$, for each proposition B in $\mathbf{A}$.

A function κ from the set of worlds W into the natural numbers N, $\kappa \colon W \to N$, is a *pointwise ranking function* on W if and only if $\kappa(w) = 0$ for at least one world w in W. Each pointwise ranking function κ on W induces a regular and completely minimitive ranking function ρ_κ on every field of propositions $\mathbf{A}$ over W by defining

7. $\rho_\kappa(A) = \min\{\kappa(w) \colon w \in A\}$ ($= \infty$ if $A = \varnothing$)

Huber (2006) discusses under which conditions a ranking function on a field of propositions **A** induces a pointwise ranking function on the underlying set of worlds *W*.

The rank of a proposition *A*, ρ(*A*), represents the degree to which an agent with ranking function ρ disbelieves *A*. If ρ(*A*) = 0, the agent does not disbelieve *A*. However, this does not mean that she believes *A*. She may well suspend judgment and neither disbelieve *A* nor its complement or negation *W\A* (in this case ρ(*A*) = ρ(*W\A*) = 0). Rather, belief in a proposition is characterized by disbelief in its negation: an agent with ranking function ρ: **A** → *N*∪{∞} *believes A* ∈ **A** if and only if ρ(*W\A*) > 0. The *belief set* Bel$_ρ$ of an agent with ranking function ρ: **A** → *N*∪{∞} is the set of all propositions she believes:

Bel$_ρ$ = {*A* ∈ **A**: ρ(*W\A*) > 0}

The axioms of ranking theory require an agent to not disbelieve both a proposition and its negation – i.e. at least one of *A*, *W\A* has to be assigned rank 0. Thus an agent with ranking function ρ: **A** → *N*∪{∞} believes *A* ∈ **A** if and only if ρ(*W\A*) > ρ(*A*). For a given ρ: **A** → *N*∪{∞}, this suggests to define the *belief function induced by* ρ, β$_ρ$: **A** → *Z*∪{±∞}, such that for all propositions *A* in **A**:

β$_ρ$(*A*) = ρ(*W\A*) − ρ(*A*)

β$_ρ$ assigns positive numbers to the propositions that are believed, negative numbers to the propositions that are disbelieved, and 0 to those propositions and their negations with respect to which the agent suspends judgment. As a consequence,

Bel$_ρ$ = {*A* ∈ **A**: β$_ρ$(*A*) > 0}

Bel$_ρ$ is consistent and deductively closed in the finite sense, for every ranking function ρ on **A**. That is, ∩**B** ≠ ∅ for every finite **B** ⊆ Bel$_ρ$; and *A* ∈ Bel$_ρ$ if there is a finite **B** ⊆ Bel$_ρ$ such that ∩**B** ⊆ *A*, for any *A* ∈ **A**. If ρ: **A** → *N*∪{∞} is countably/completely minimitive, Bel$_ρ$ is consistent and deductively closed in the following countable/complete sense: ∩**B** ≠ ∅ for every countable/arbitrary **B** ⊆ Bel$_ρ$; and *A* ∈ Bel$_ρ$ if there is a countable/arbitrary **B** ⊆ Bel$_ρ$ such that ∩**B** ⊆ *A*, for any *A* ∈ **A**. As will be seen below, from a diachronic

point of view the converse is true as well. However, first we have to discuss how an epistemic agent is to update her ranking function when she learns new information.

UPDATE RULES

A theory of epistemic states is incomplete if it does not account for the way the epistemic states are updated when the agent receives new information. As there are different formats in which the agent may receive new information, there are different update rules. The simplest and most unrealistic case is that of the agent becoming certain of a new proposition. This case is covered by

Plain Conditionalization

If the agent's epistemic state at time t is represented by the ranking function ρ *on* **A***, and if, between t and t', the agent becomes certain of the proposition E* ∈ **A** *and of no logically stronger proposition E+* ⊂ *E, E+* ∈ **A***, then the agent's epistemic state at time t' should be represented by the ranking function* ρ' = ρ(·|*E*) *on* **A***.*

We usually do not learn by becoming certain of a proposition, though. In most cases the new information merely changes the strength of our beliefs in various propositions. This is illustrated by a variation of an example due to Jeffrey (1983). Let our agent be interested in the color of the carpet of her hotel room. At time *t*, before checking in, she neither believes nor disbelieves any of the following three hypotheses: the carpet is beige (*beige*), the carpet is brown (*brown*), the carpet is black (*black*). However, she is certain that the carpet is either beige or brown or black. The relevant part of her ranking function at time *t* thus looks as follows: ρ(*beige*) = ρ(*not beige*) = ρ(*brown*) = ρ(*not brown*) = ρ(*black*) = ρ(*not black*) = ρ(*beige or brown or black*) = 0, ρ(*neither beige nor brown nor black*) = ∞.

At time *t'*, after checking in and when opening the door to her room, it appears to the agent that the carpet is rather dark. As a consequence she now believes that the carpet is either brown or black. But since it is late at night, the curtains are closed, and she has not turned on the light yet, she cannot tell whether the carpet is brown or black. Her ranks for the relevant propositions

thus change to the following values: ρ'(*beige*) = ρ'(*not brown*) = ρ'(*not black*) = 1, ρ'(*not beige*) = ρ'(*brown*) = ρ'(*black*) = 0.

A change in the strength of the agent's beliefs about the color of the carpet will affect the strength of her beliefs about the color of, say, the furniture in the hotel room. For instance, at time *t*, our agent is pretty confident that the hotel room does not have dark furniture if the carpet is brown – and similarly if the carpet is black. She is also pretty confident that the hotel room has dark furniture if the carpet is beige. The relevant part of her ranking function at time *t* looks as follows: ρ(*dark|brown*) = ρ(*dark|black*) = 3, ρ(*dark|beige*) = 0. This implies that, at time *t*, the agent neither believes the furniture is dark nor that it is not dark, ρ(*dark*) = ρ(*not dark*) = 0.

The important question now is how the agent should update the rest of her ranking function (including the ranks for the propositions about the color of the furniture) when her ranks for the propositions about the color of the carpet change as specified above. The answer, already formulated in Spohn (1988), is given by

Spohn Conditionalization

If the agent's epistemic state at time t is represented by the ranking function ρ *on* **A***, and if, between t and t', the agent's ranks on the partition {E_i \in \begin{cal}A\ end{cal}: E_i \in I} change to n_i \in N\cup{\infty} with min_i{n_i}=0 (n_i=0 if E_i=W and n_i=\infty if E_i=\emptyset), and the agent's finite ranks change on no finer partition, then the agent's epistemic state at time t' should be represented by the ranking function* ρ*' = min{*ρ(·|E_1) + r_1*, ...,* ρ(·|E_n) + r_n*, ...} on* **A***.*

Applied to our example this means that, at time *t'*, the agent's rank for the proposition that the furniture is dark should be ρ'(*dark*) = min{ρ(*dark|beige*) + 1, ρ(*dark|brown*) + 0, ρ(*dark|black*) + 0} = 1. That is, at time *t'*, the agent believes, if only very weakly, that the furniture is not dark.

Spohn Conditionalization covers Plain Conditionalization as a special case. Shenoy (1991) presents an update rule for evidence of a still different format.

JUSTIFICATION

Ranking theory tells an epistemic agent how to organize her beliefs, and how to update her beliefs when she receives new information of various formats. Why should the agent follow those prescriptions?

The answer to this question requires a bit of terminology. An agent's *degree of entrenchment* for the proposition *A* is the number of information sources providing the information *A* that it takes for the agent to give up her disbelief in *A*. If the agent does not disbelieve *A* to begin with, her degree of entrenchment for *A* is 0. If no finite number of information sources providing the information *A* makes the agent give up her disbelief in *A*, her degree of entrenchment for *A* is ∞.

Degrees of entrenchment are used to measure an epistemic agent's degrees of disbelief. If you want to measure my degree of disbelief for the proposition that Madrid is the capitol of Spain, you put me on a busy plaza in the center of Madrid and count the number of people passing by and telling me that Madrid is the capitol of Spain. My degree of entrenchment for the proposition that Madrid is the capitol of Spain equals *n* just in case I stop disbelieving that Madrid is the capitol of Spain after *n* people have passed by and told me it is – provided all those people are independent and equally reliable, indeed minimally positively reliable. Most people (and certainly all people in Madrid) are more than minimally positively reliable, though. An agent's *degree of disbelief* in *A* is therefore defined as the number of information sources providing the information *A* that it would take for the agent to give up her disbelief that *A* if those information sources were independent and minimally positively reliable.

Now we can explain why an agent's degrees of disbelief should obey the ranking calculus and thus be ranks, and why she should update her ranks according to Spohn Conditionalization. She should do so because doing so is necessary and sufficient for her to always have consistent and deductively closed beliefs. More precisely, Huber (2007) proves the following.

Consistency Theorem

An agent's belief set is and will always be consistent and deductively closed in the finite/countable/complete sense (and possibly conditional on some evidential proposition) if and only if this agent's degree of disbelief function is a finitely/countably/completely minimitive

ranking function and the agent updates according to Plain/Spohn/Shenoy Conditionalization when she receives information of the appropriate format.

Seen this way, the axioms and update rules of ranking theory are nothing but a diachronic version of consistency and deductive closure.

FUTURE TRENDS

One question in artificial intelligence is how an agent should update her epistemic state if she learns new conceptual information without also learning anything factual about the world she lives in. There are several ways in which such a conceptual change may occur. The agent may learn a new concept as when an enological ignoramus learns the concept *barrique*. Or the agent may learn that she has omitted a possibility from her set of worlds as when an enological ignoramus learns that there are rosé wines besides red and white wines. All these conceptual changes involve the adoption of a new set of worlds *W* and, consequently, a new field of propositions **A** on the side of the agent. None of these conceptual changes seems to be adequately modeled by any of the formalisms mentioned at the beginning. Ranking theory is able to adequately model those conceptual changes by employing the so called *ur* or *tabula rasa* ranking – i.e. that ranking function that assigns rank 0 to every proposition. If the agent adds new possibilities to her set of worlds she should simply assign rank 0 to all those new possibilities. Similarly in case the agent replaces the old worlds by richer worlds. Huber (2009) discusses this and other future trends.

CONCLUSION

Ranking functions are an indispensable tool for artificial intelligence. First, they seem to adequately model most if not all of those phenomena that are dealt with in both qualitative as well as quantitative approaches to uncertainty. Second, they provide a link between these two classes of approaches that has been missing so far. Third, they can deal with phenomena that neither qualitative nor quantitative approaches seem to be able to deal with.

REFERENCES

Alchourrón, C.E. & Gärdenfors, P. & Makinson, D. (1985), On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic* 5**0**, 510-530.

Dempster, A.P. (1968), A Generalization of Bayesian Inference. *Journal of the Royal Statistical Society*. Series B (Methodological) **30**, 205-247.

Dubois, D. & Prade, H. (1988), *Possibility Theory. An Approach to Computerized Processing of Uncertainty*. New York: Plenum Press.

Gärdenfors, P. (1988), *Knowledge in Flux. Modeling the Dynamics of Epistemic States*. Cambridge, MA: MIT Press.

Halpern, J.Y. (2003), *Reasoning About Uncertainty*. Cambridge, MA: MIT Press.

Halpern, J.Y. & Fagin, R. & Moses, Y. & Vardi, M.Y. (1995), *Reasoning About Knowledge*. Cambridge, MA: MIT Press.

Hintikka, J. (1961), *Knowledge and Belief. An Introduction to the Logic of the Two Notions*. Ithaca, NY: Cornell University Press.

Huber, F. (2006), Ranking Functions and Rankings on Languages. *Artificial Intelligence 170*, 462-471.

Huber, F. (2007), The Consistency Argument for Ranking Functions. *Studia Logica 86*, 299-329.

Huber, F. (2009), Belief and Degrees of Belief. In F. Huber & C. Schmidt-Petri (eds.), \textit{Degrees of Belief}. Berlin: Springer.

Jeffrey, R.C. (1983), *The Logic of Decision*. 2nd ed. Chicago: University of Chicago Press.

Jeffrey, R.C. (2004), *Subjective Probability: The Real Thing*. Cambridge: Cambridge University Press.

Kraus, S. & Lehmann, D. & Magidor, M. (1990), Non-monotonic Reasoning, Preferential Models, and Cumulative Logics. *Artificial Intelligence 40*, 167-207.

Makinson, D. (2005), *Bridges from Classical to Non-monotonic Logic*. London: College Publications.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco: Morgan Kaufmann.

Rott, H. (2001), *Change, Choice, and Inference. A Study of Belief Revision and Nonmonotonic Reasoning*. Oxford: Oxford University Press.

Shafer, G. (1976), *A Mathematical Theory of Evidence*. Princteon, NJ: Princeton University Press.

Shenoy, P.P. (1991), On Spohn's Rule for Revision of Beliefs. *International Journal of Approximate Reasoning* **5**, 149-181.

Spohn, W. (1988), Ordinal Conditional Functions: A Dynamic Theory of Epistemic States. In W.L. Harper & B. Skyrms (eds.), *Causation in Decision, Belief Change, and Statistics II*. Dordrecht: Kluwer, 105-134.

Spohn, W. (1990), A General Non-Probabilistic Theory of Inductive Reasoning. In R.D. Shachter & T.S. Levitt & J. Lemmer & L.N. Kanal (eds.), *Uncertainty in Artificial Intelligence* **4**. Amsterdam: North-Holland, 149-158.

Spohn, W. (manuscript), *Ranking Theory*.

KEY TERMS

Belief: An agent with ranking function ρ: $\mathbf{A} \rightarrow N \cup \{\infty\}$ *believes* A if and only if $\rho(W \backslash A) > 0$ – equivalently, if and only if $\rho(W \backslash A) > \rho(A)$.

Belief Set: The *belief set* of an agent with ranking function ρ: $\mathbf{A} \rightarrow N \cup \{\infty\}$ is the set of propositions the agent believes, $\mathrm{Bel}_\rho = \{A \in \mathbf{A}: \rho(W \backslash A) > 0\}$.

Conditional Ranking Function: The *conditional ranking function* $\rho(\cdot|\cdot)$: $\mathbf{A} \times \mathbf{A} \rightarrow N \cup \{\infty\}$ based on the ranking function ρ on $\mathbf{A}$ is defined such that for all propositions A, B in $\mathbf{A}$: $\rho(A|B) = \rho(A \cap B) - \rho(B)$ if $A \neq \varnothing$, and $\rho(\varnothing|B) = \infty$.

R

Completely Minimitive Ranking Function: A ranking function ρ on a γ-field of propositions $\mathbf{A}$ is *completely minimitive* if and only if $\rho(\cup \mathbf{B}) = \min\{\rho(A): A \in \mathbf{B}\}$ for each set of propositions $\mathbf{B} \subseteq \mathbf{A}$.

Countably Minimitive Ranking Function: A ranking function ρ on a σ-field of propositions $\mathbf{A}$ is *countably minimitive* if and only if $\rho(A_1 \cup \ldots \cup A_n \cup)$ $= \min\{\rho(A_1), \ldots, \rho(A_n), \ldots\}$ for all propositions $A_1 \in \mathbf{A}, \ldots A_n \in \mathbf{A}, \ldots$

Pointwise Ranking Function: A function κ from the set of worlds W into the natural numbers N, κ: $W \rightarrow N$, is a *pointwise ranking function* on W if and only if $\kappa(w) = 0$ for at least one world w in W.

Degree of Disbelief: An agent's *degree of disbelief* in the proposition A is the number of information sources providing the information A that it would take for the agent to give up her disbelief that A if those information sources were independent and minimally positively reliable.

Degree of Entrenchment: An agent's *degree of entrenchment* for the proposition A is the number of information sources providing the information A that it takes for the agent to give up her disbelief in A.

Ranking Function: A function ρ on a field of propositions $\mathbf{A}$ over a set of worlds W into the natural numbers extended by ∞, ρ: $\mathbf{A} \rightarrow N \cup \{\infty\}$, is a (*finitely minimitive*) *ranking function* on $\mathbf{A}$ if and only if for all propositions A, B in $\mathbf{A}$: $\rho(W) = 0$, $\rho(\varnothing) = \infty$, $\rho(A \cup B) = \min\{\rho(A), \rho(B)\}$.

RBF Networks for Power System Topology Verification

Robert Lukomski
Wroclaw University of Technology, Poland

Kazimierz Wilkosz
Wroclaw University of Technology, Poland

INTRODUCTION

A necessary condition for monitoring and control of a Power System (PS) is possessing a credible model of this system. The PS model for a need of dispatchers in national control centre is created in real time. An important element of such a model is a topology model. PS Topology Verification (PSTV) is an important problem in PS engineering. Often this problem is solved together with PS state estimation (Lukomski, & Wilkosz, 2000; Mai, Lefebvre, & Xuan, 2003). Methods, that enable such a solution of the problem, are sophisticated and usually time consuming. They require successful state estimation performance but convergence problems may occur in the case of certain Topology Errors (TEs). Thus, a robust method for PSTV before a state estimation is desired.

BACKGROUND

Now, the growth rate of Artificial Neural Networks (ANNs) application in some PS subjects is observed (Haque, & Kashtiban, 2005). One of such a subject is PSTV. It can be considered as a pattern recognition problem and then also utilization of ANN technique for solution of PSTV can be taken into account (Alves da Silva, & Quintana, 1995; Souza, Leite da Silva, & Alves da Silva, 1996, 1997, 1998). There are many references in which PSTV with use of ANNs is described. In (Tian, Zhu & Zhang, 1995) use of ANN as a part of an expert system to rule extraction is presented. One of the first method for such PSTV has assumed utilization of one ANN for whole PS (Vinod Kumar, Srivastava, Shah, & Mathur, 1996). In the case of this method the complexity of the ANN structure grows rapidly with the size of a power network. There are the problems with learning and classification process in a case of large ANNs. In other attempts to solve the problem of PSTV with use of ANNs one can observe utilization of additional knowledge on PS (Garcia-Lagos, Joya, Marin, & Sandoval, 2003; Delimar, Hebel, & Pavić, 2001, 2002, 2003a, 2003b). Such approach allows reducing size of utilized ANNs. The learning and classification process become more effective and the verification method is more efficient. The considered approach is also utilized in the case of the method, which is further presented.

DESCRIPTION OF THE CONSIDERED SOLUTION

To ensure that in the described method a larger knowledge on PS will be utilized than it is in other methods for PSTV, so-called unbalance indices are introduced. Taking into account the nature of the solved problem and to accomplish the best features of the PSTV, Radial Basis Function Networks (RBFNs) are utilized.

Power System Model

Elements of the PS topology model are nodes (representing electrical nodes) and branches (representing power lines, transformers, loads etc.). The assumption, that every branch in a PS model is modeled as the π-equivalent circuit (Fig. 1), is adopted. It is assumed that there is an accessible credible measurement data set of such quantities as: active and reactive power flows at the ends of each branch, power injections, loads and voltage magnitudes at each node. Usually, if a branch is not included in PS model the measurement data related to the branch are not taken into account in carried out analyses.

Figure 1. The assumed π model of the branch, $\mathbf{Z}_{kl} = R_{kl} + j X_{kl}$, $\mathbf{Y}_k = jB_{kl}$, $\mathbf{Y}_l = jB_{lk}$, $B_{kl} = B_{lk} = B$. B is a half of the capacitive susceptance of the branch.
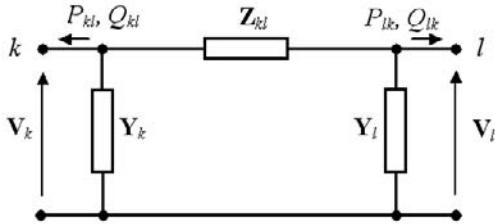
Unbalance Indices

Using Kirchhoff's and Ohm's Laws, PS can be described by many relationships among measured quantities. If there are no TEs, all these relationships are fulfilled. When TE occurs some of the relationships become unfulfilled. It should be underlined that if a branch is not included in the PS model, the relationships for this branch are not considered, because measurement data for it are not taken into account. In the described approach to have possibility of examination of relationships for all nodes and all branches independently of their correct or incorrect inclusion in the PS model the so-called unbalance indices for nodes and branches are introduced (Lukomski, 2002). These indices are shown in Table 1.

It should be noted that the nodal unbalance indices instead of power flow measurement data are taken into account when branch unbalance indices are calculated. This fact allows considering branch unbalance indices independently of correct or incorrect inclusion of branches in the PS model.

Unbalance indices create characteristic sets of values for different cases of modeling PS. If the topology model is correct and there are no errors burdening measurement data, all nodal unbalance indices are equal to zero and branch unbalance indices are near to zero, as well. The same situation is, when there is a branch that is actually out of operation but it is included in the topology model (the inclusion error). If a branch is actually in operation in PS but it is not included in the topology model (the exclusion error), then; (i) the unbalance indices for terminal nodes of this branch considerably differ from zero, (ii) the unbalance indices for the considered branch are equal to zero, (iii) absolute values of the unbalance indices for other branches, that are incident to the nodes mentioned under (i), have especially large values.

It should be stressed that the behavior of unbalance indices for active power and for reactive power is the same for the same TE.

Analyzing unbalance indices for nodes and branches one can observe that the exclusion error of the branch j has no influence on: (i) unbalance indices for nodes, that are not terminal nodes of the branch j, (ii) unbalance indices for branches that are not incident to the

Table 1. Active and reactive power unbalance indices for nodes and branches

	Node	Branch
Active power	$W_{Pk} = \sum_{i \in I_k} P_{ki}$	$W_{Pkl} = -W_{Pk} - W_{Pl} + R_{kl}W$
Reactive power	$W_{Qk} = \sum_{i \in I_k} Q_{ki}$	$W_{Qkl} = -W_{Qk} - W_{Ql} + X_{kl}W - B_{kl}\left(V_k^2 + V_l^2\right)$
Description: W_{Pk}, W_{Qk} – unbalance indices for the node k for active and reactive power respectively; W_{Pkl}, W_{Qkl} – unbalance indices for the branch connecting the nodes k and l for active and reactive power respectively; I_k - a set of the nodes connected to the node k; P_{ki}, Q_{ki} – active and reactive power flows in the branch connecting the nodes k and i at the node k; R_{kl}, X_{kl}, $B_{kl} - \pi$ model parameters for the branch connecting the nodes k and l (Fig. 1); V_k, V_l – voltage magnitudes at the nodes k and l respectively; $$W = \frac{W_{Pk}^2 + \left(W_{Qk} + B_{kl}V_k^2\right)^2}{V_k^2}.$$		

terminal nodes of the branch *j*. This observation shows existence of the local effect of TE. In this situation one can conclude about correctness of modeling the distinguished branch *j* on the basis of investigations of unbalance indices for certain areas of the power network: A_j^k, A_j^l, where: *k, l* are numbers of the terminal nodes of the branch *j*. A_j^x $x \in \{k, l\}$ is the area, in which the branch *j* exists with the central node *x*. The area A_j^x comprises: (i) the node *x* (being one of the terminal nodes of the branch *j*), (ii) the branch *j* and all other branches incident to the node *x*, (iii) all nodes which are connected with the node *x* by the branches mentioned under (ii).

The Need of Use of ANNs

The earlier considerations regarding the unbalance indices pertain to the ideal situation. In real situations, measurement data are burdened with errors and also one can occur multiple TEs. In such situations, the earlier-described effects of TEs, effects of occurrence of measurement errors and TEs other than the one, that is incorrect modeling the considered branch, overlap each other. In real situations, the problem of PSTV is a complex problem (Lukomski, 2002). Taking into account the analysis of the behavior of the unbalance indices, one can state that in the described situation the problem of PSTV can be treated as the problem of pattern recognition and then utilization of ANNs can be considered as a proper idea of the solution of the PSTV problem.

On the basis of the earlier considerations it can be stated that the whole PSTV process can be decomposed into many simpler PSTV processes. One such process can be limited to the area A_j^x. If one assumes utilization of ANNs then for each of the distinguished processes the separate ANN should be constructed. Possibly simple and fast learning ANNs is desired and therefore attention has been paid to RBFNs (Meireles, Almeida, & Simões, 2003).

Principle of the Method

The proposed topology verification method consists of the following steps:

• calculation of unbalance indices for nodes and branches,

• pre-processing of the unbalance indices (pre-processing standardization),
• local classification,
• global classification.

The Pre-Processing Standardization

The pre-processing standardization of each unbalance index is realized using Radial Basis Function (RBF) unit with Gaussian transfer function:

$$f(w) = \exp\left(-\frac{w^2}{2\sigma^2}\right), \tag{1}$$

where: *w* is a value of the considered unbalance index, σ is the width parameter.

If an unbalance index is close to zero, the RBF unit output is close to one. If an unbalance index is significantly different from zero, the RBF unit output is close to zero. The pre-processing standardization allows keeping input values for local classifiers (in the next step of the method) in the range (0; 1]. The σ parameter for the index W_{Pk} is calculated as follows (errors are assumed to be independent):

$$\sigma_{WPk}^2 = a \sum_{l \in I_k} \sigma_{Pkl}^2, \tag{2}$$

where σ_{Pkl} is a standard deviation of data of the active power flow P_{kl}; *a* is a correction coefficient selected in an experimental way.

The width parameter for the branch unbalance index W_{Pkl} is given by:

$$\sigma_{WPkl}^2 = a(\sigma_{WPk}^2 + \sigma_{WPl}^2), \tag{3}$$

Width parameters for unbalance indices for reactive power are calculated in the similar way. One has assumed a = 2 for active power unbalance indices and a = 1,8 for reactive power unbalance indices.

The Local Classification

The purpose of the considered step of the method is classification of correctness of modeling branches of PS. During the described step the local effect of TEs is taken into account.

Each local classifier is RBFN. One local classifier corresponds to one node of a considered power network.

If the considered node has the number k then inputs for a local classifier, that corresponds to this node, are the results of the pre-processing of active and reactive power unbalance indices for: (i) the node k, (ii) the nodes having numbers from the set I_k, (iii) each branch connecting the node k and the node l, under assumption $l \in I_k$. The number of outputs of a local classifier is equal to the number of branches connecting the node k with the nodes having numbers from the set I_k. The criterion for taking a decision on correctness of modeling a branch is as follows:

$$D_l = \begin{cases} \text{the branch } l \text{ is incorrectly modelled} & \text{when} \quad Y_l \le -0.5 \\ \text{the neutral decision} & \text{when} \quad Y_l \in (-0.5, 0.5) \\ \text{the branch } l \text{ is correctly modelled} & \text{when} \quad Y_l \ge 0.5 \end{cases}$$

(4)

where: D_l is a decision, Y_l is an output value corresponding to the branch between the node k and the node l.

The Global Classification

The global decision unit processes decisions of the local classifications and produces final decisions on correctness of modeling branches of PS. To take a final decision on correctness of modeling a selected branch the outputs of two local classifiers are considered. These classifiers corresponding to the terminal nodes of the considered branch. If decisions of local classifiers are different and none of them is the neutral decision or each of the local classifiers produces the neutral decision then the final decision is the neutral one. In other cases the final decision is different from the neutral one.

Computational Example

The presented method was implemented in the MATLAB environment. The method has been tested using the IEEE 14-bus test system (Fig. 2). It has been assumed that: (i) all branches are actually in operation, (ii) single and multiple TEs are considered, (iii) measurement data are burdened with small errors (Gaussian noise), (iv) wide range of load curve changes is taken into account.

Learning of each local classifier (being a RBFN) has been performed using Orthogonal Least Squares (OLS) algorithm (Chen, Cowan & Grant, 1991). Learning sets (200 – 400 learning patterns) were created separately for each RBFN. They comprised results of pre-processing unbalance indices and appropriate verification decisions. Learning of a RBFN was stopped when Sum Square Error (SSE) achieved the 10^{-4} level.

For the distinguished RBFN the number of hidden units depends on a number of branches incident to the appropriate node. For the particular nodes of the test system many different topologies of RBFNs were

Figure 2. The IEEE-14 bus test system
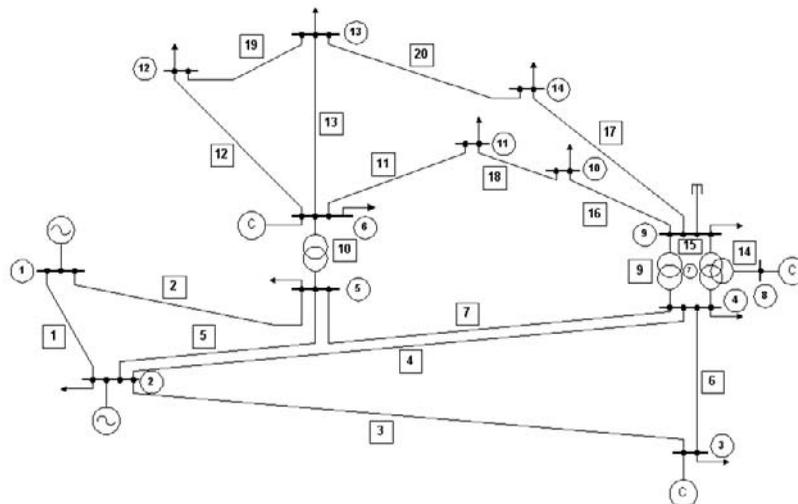
trained and tested. The characteristics of the local classifiers having the best performance is presented in Table 2.

Testing the local classifiers has been performed with use of the test set having about 2000 patterns that had not been used in the training phase. The cases with single and double TEs were considered. In the cases with single TEs only the correct decisions were produced. In the cases with double TEs the correct and neutral decisions were observed.

Table 3 shows a probability of taking the neutral decision p_n in the verification process for the different branches of the test system when there are double TEs. During the test stage, some doubtful cases have occurred and the neutral decisions have been taken for the branches with numbers 19 and 20. In these cases there has been no possibility to state the correctness of the considered branch in the test system. A reason was relatively small level of power flows in the mentioned branches. The obtained results show that the efficiency of the RBF classifiers is very high.

FUTURE TRENDS

Utilization of ANNs to handle the problem of PSTV seems to be very promising. However, the up-to-date methods do not give satisfying results in all possible real cases. The analyses have revealed that the application of pure neural models is not too effective. Utilization of ANNs and additional knowledge on PSs can result in much more efficient solutions. Also, it should be stressed that combining various artificial intelligence techniques can give interesting solutions from the view point of efficiency and performance time of a PSTV process.

CONCLUSION

The presented method allows performing PSTV independently of state estimation. It combines knowledge on PS and utilization of RBFNs. It utilizes the local effect of TEs. The whole PSTV process comprises many local processes realized by use of the classifiers assigned to the nodes of a power network. It makes possible to avoid constructing a large and complex ANN for a whole power network, as it is made in (Vinod Kumar, Srivastava, Shah, & Mathur, 1996).

Table 2. Characteristics of the local RBF classifiers corresponding to the nodes of the IEEE 14-bus test system. N – the number of the node, N_{inp} – the number of inputs, N_{hu} – the number of hidden units, N_{out} - number of outputs.

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14
N_{inp}	10	18	10	22	18	18	14	6	18	10	10	10	14	10
N_{hu}	20	51	30	98	76	47	38	4	62	18	15	20	25	23
N_{out}	2	4	2	5	4	4	3	2	4	2	2	2	3	2

Table 3. A probability of taking the neutral decision p_n in the verification process for the different branches of the test system when there are double topology errors.

j	1 - 18	19	20
p_n	0.00	0.11	0.10

Taking into account the decomposition of the PSTV process, the described method is close to the method from (Garcia-Lagos, Joya, Marin & Sandoval, 1998, 2003) and also to the method from (Delimar, Hebel & Pavić, 2001, 2002, 2003a, 2003b). However, the characterized method utilizes larger knowledge on PS than the method from (Garcia-Lagos, Joya, Marin & Sandoval, 2003) or the method from (Delimar, Hebel & Pavić, 2001, 2002, 2003a, 2003b). A consequence of this fact is decreasing sizes of ANNs of which utilization is assumed by the here-considered method in comparison with the method from (Garcia-Lagos, Joya, Marin & Sandoval, 2003) or the method from (Delimar, Hebel & Pavić, 2001, 2002, 2003a, 2003b).

The method assumes that the local classifiers are RBFNs. Their learning process is relatively short, comparing with multilayer feedforward neural networks. Using the OLS algorithm gives fast learning convergence. However, it should be stressed that RBFNs have much higher number of hidden units in comparison with multilayer feedforward neural networks.

The described method is capable to handle single and multiple TEs. It allows performing very efficient PSTV. Another advantage of the method is low sensitivity of the PSTV process quality to changes of PSs load curve.

REFERENCES

Alves da Silva, A. P., & Quintana, V. H. (1995). Pattern Analysis in Power System State Estimation. *Electrical Power and Energy Systems*. 17(1), 51–60.

Chen, S., Cowan, C. B., & Grant, P. M. (1991). Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*. 2(2), 302–309.

Delimar, M., Hebel, Z., & Pavić, I. (2001). Power system topology identification using neural networks. Part I – line processing. *The IASTED International Conference Power and Energy Systems*, Clearwater, Florida, USA. 129–133.

Delimar, M., Hebel, Z., & Pavić, I. (2002),. Power System Topology Identification Using Neural Networks. Part II-Node Processing. *The IASTED International Conference Power and Energy Systems*, Crete, Greece. 67–71.

Delimar, M., Pavić, I., & Hebel, Z. (2003a). Power System Topology Identification Using Neural Networks. Part III-Unsupervised parts of the power system. *The 7th IASTED International Multi-Conference Power and Energy Systems*, Palm Springs, CA, USA. 83-87.

Delimar, M., Pavić, I., & Hebel, Z. (2003b). Artificial neural networks in power system topology recognition. *EUROCON*. 2, 287–291.

Garcia-Lagos, F., Joya, G., Marin, F. J., Sandoval, F. (2003). A modular power system topology assessment based on Gaussian potential functions. *IEE Proceedings on Generation, Transmission and Distribution*. 150(5), 635–640.

Haque, M.T., & Kashtiban, A.M. (2005). Application of neural networks in power systems; a review. *Transactions on Engineering, Computing and Technology*. 14(6), 53-57.

Lukomski, R. (2002). New Approach to Power System Topology Verification. *The International Conference on Modern Electric Power Systems*, Wroclaw, Poland. 517–521.

Lukomski, R., & Wilkosz, K. (2000). Power System Topology Verification: Assessment of Different Approaches. *The 4-th International Conference on Control of Power Systems*, Bratislava, Slovakia. 269-274.

Mai, H.V., Lefebvre, S., & Xuan, D.D. (2003). A review of methods for topology errors detection. *IEEE PES Transmission and Distribution Conference and Exposition*. 1, 144-149.

Meireles, M.R.G., Almeida, P.E.M., & Simões, M.G. (2003). A Comprehensive Review for Industrial Applicability of Artificial Neural Networks. *IEEE Transactions on Industrial Electronics*. 50(3), 585-601.

Souza, J. C. S., Leite da Silva, A. M., & Alves da Silva, A. P. (1996). Data Debugging for Real-Time Power System Monitoring Based on Pattern Analysis. *IEEE Transactions on Power Systems*. 11(3), 1592 – 1599.

Souza, J. C. S., Leite da Silva, A. M., & Alves da Silva, A. P. (1997). Data Visualization and Identification of Anomalies in Power System State Estimation Using Artificial Neural Networks. *IEE Proceedings on Generation Transmission and Distribution*. 144(5), 445–455.

Souza J. C. S., Leite da Silva A. M., & Alves da Silva A. P. (1998). Online topology determination and bad data suppression in power system operation using artificial neural networks. *IEEE Transactions on Power Systems*. 13(3), 796–803.

Tian T, Zhu M., & Zhang B. (1995). An Artificial Neural Network-Based Expert System for Network Topological Error Identification. *The IEEE International Conference on Neural Networks*, Perth, WA. 2, 882–886.

Vinod Kumar D. M., Srivastava S. C., Shah S., & Mathur S. (1996). Topology Processing and Static State Estimation Using Artificial Neural Networks. *IEE Proceedings on Generation, Transmission and Distribution*. 143(1), 99–105.

KEY TERMS

Neutral Decision: In fact, the lack of any decision.

Orthogonal Least Squares (OLS) Algorithm: Algorithm describing a Gram-Schmidt orthogonalisation process which ensures that each new column added to the result matrix of the growing subset is orthogonal to all previous columns. This considerably simplifies the equation for the change in learning error and results in a more efficient algorithm.

Power System State Estimation: A process, which leads to calculation of a power system state vector using incoming measurement data and a mathematical power system model. A power system state vector fully specifies any state in which a power system can be.

Power System Topology Error: Inconsistency among the real power network connectivity and the power system topology model.

Power System Topology Model: A description of the physical connections in a power system.

Power System Topology Verification: Proving or disproving the correctness of a power system topology model.

Radial Basis Function Network: A type of artificial neural network which uses radial basis functions as activation functions. Typically, it consists of one hidden layer of Radial Basis Function (RBF) neurons (units). RBF hidden layer units have a receptive field which has a centre: that is, a particular input value at which they have a maximal output. Their output tails off as the input moves away from this point. Generally, the hidden unit function is a Gaussian. They are used in classification and approximation problems.

Unbalance Index: The left-hand side of the appropriate relationship, considered in the form in which its right-hand side is equal to zero. The mentioned relationship is a balance of active (reactive) powers at a node or a relationship among active (reactive) power flows at the ends of a branch.

Representing Non–Rigid Objects with Neural Networks

José García-Rodríguez
University of Alicante, Spain

Francisco Flórez-Revuelta
University of Alicante, Spain

Juan Manuel García-Chamizo
University of Alicante, Spain

INTRODUCTION

Self-organising neural networks try to preserve the topology of an input space by means of their competitive learning. This capacity has been used, among others, for the representation of objects and their motion. In this work we use a kind of self-organising network, the Growing Neural Gas, to represent deformations in objects along a sequence of images. As a result of an adaptive process the objects are represented by a topology representing graph that constitutes an induced Delaunay triangulation of their shapes. These maps adapt the changes in the objects topology without reset the learning process.

BACKGROUND

Self-organising maps, by means of a competitive learning, make an adaptation of the reference vectors of the neurons, as well as, of the interconnection network

among them; obtaining a mapping that tries to preserve the topology of an input space. Besides, they are able of a continuous re-adaptation process even if new patterns are entered, with no need to reset the learning.

These capacities have been used for the representation of objects (Flórez, García, García & Hernández, 2001)] (Figure 1) and their motion (Flórez, García, García & Hernández, 2002) by means of the Growing Neural Gas (GNG) (Fritzke, 1995) that has a learning process more flexible than other self-organising models, like Kohonen maps (Kohonen, 2001).

These two applications, representation of objects and their motion, have in many cases temporal constraints, reason why it is interesting the acceleration of the learning process. In computer vision applications the condition of finalization for the GNG algorithm is commonly defined by the insertion of a predefined number of neurons. The election of this number can affect the quality of the adaptation, measured as the topology preservation of the input space (Martinetz & Schulten, 1994).

Figure 1. Representation of two-dimensional objects with a self-organising network
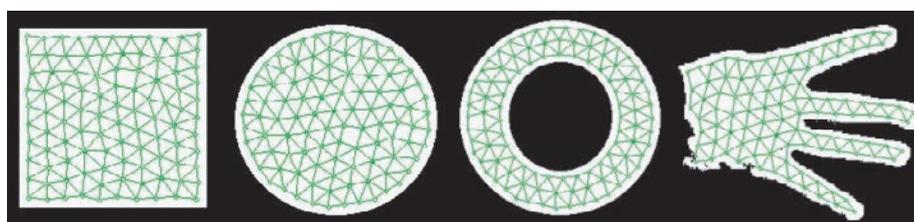
In this work GNG has been used to represent two-dimensional objects shape deformations in sequences of images, obtaining a topology representing graph that can be used for multiple tasks like representation, classification or tracking. When deformations in objects topology are small and gradual between consecutive frames in a sequence of images, we can use previous maps information to place the neurons without reset the learning process. Using this feature of GNG we achieve a high acceleration of the representation process.

One way of selecting points of interest in 2D shapes is to use a topographic mapping where a low dimensional map is fitted to the high dimensional manifold of the shape, whilst preserving the topographic structure of the data. A common way to achieve this is by using self-organising neural networks where input patterns are projected onto a network of neural units such that similar patterns are projected onto units adjacent in the network and vice versa. As a result of this mapping a representation of the input patterns is achieved that in post-processing stages allows one to exploit the similarity relations of the input patterns. Such models have been successfully used in applications such as speech processing (Kohonen, 2001), robotics (Ritter & Schulten, 1986), (Martinez, Ritter, & Schulten, 1990) and image processing (Nasrabati & Feng, 1988). However, most common approaches are not able to provide good neighborhood and topology preservation if the logical structure of the input pattern is not known a priori. In fact, the most common approaches specify in advance the number of neurons in the network and a graph that represents topological relationships between them, for example, a two-dimensional grid, and seek the best match to the given input pattern manifold. When this is not the case the networks fail to provide good topology preserving as for example in the case of Kohonen's algorithm.

REPRESENTATION AND TRACKING OF NON-RIGID OBJECTS WITH TOPOLOGY PRESERVING NEURAL NETWORKS

This section is organized as follows: first we provide a detailed description of the topology learning algorithm GNG. Next an explanation on how GNG can be applied to represent objects that change their shapes in a sequence of images is given. And finally a set of experimental results using GNG to represent different input spaces is presented in.

The approach presented in this paper is based on self-organising networks trained using the Growing Neural Gas learning method (Fritzke, 1995), an incremental training algorithm. The links between the units in the network are established through competitive hebbian learning (Martinetz, 1994). As a result the algorithm can be used in cases where the topological structure of the input pattern is not known a priori and yields topology preserving maps of feature manifold (Martinetz & Schulten, 1994).

Recent studies has presented some modifications of the original GNG algorithm to improve the robustness of the cluster analysis (Cselényi, 2005), (Cheng & Zell, 2000), (Qin & Suganthan, 2004), (Toshihiko, Iwasaki & Sato, 2003), but none of them use the structure of the map as starting point to represent deformations in a sequence of objects shapes.

Growing Neural Gas

With Growing Neural Gas (GNG) (Fritzke, 1995) a growth process takes place from a minimal network size and new units are inserted successively using a particular type of vector quantisation (Kohonen, 2001). To determine where to insert new units, local error measures are gathered during the adaptation process and each new unit is inserted near the unit which has the highest accumulated error. At each adaptation step a connection between the winner and the second-nearest unit is created as dictated by the competitive hebbian learning algorithm. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology based on some measure. Also the ending condition could it be the insertion of a predefined number of neurons or a temporal constrain. In addition, in GNG networks learning parameters are constant in time, in contrast to other methods whose learning is based on decaying parameters.

In the remaining of this Section we describe the growing neural gas algorithm and ending condition as used in this work. The network is specified as:

A set N of nodes (neurons). Each neuron $c \in N$ has its associated reference vector $w_c \in R^d$. The reference vectors can be regarded as positions in the input space of their corresponding neurons.

A set of edges (connections) between pairs of neurons. These connections are not weighted and its purpose is to define the topological structure. An edge aging scheme is used to remove connections that are invalid due to the motion of the neuron during the adaptation process.

The GNG learning algorithm to approach the network to the input manifold is as follows:

1. Start with two neurons a and b at random positions w_a and w_b in R^d.
2. Generate a random input pattern ξ according to the data distribution $P(\xi)$ of each input pattern. In our case since the input space is 2D, the input pattern is the (x,y) coordinate of the points belonging to the object. Typically, for the training of the network we generate 1000 to 10000 input patterns depending on the complexity of the input space.
3. Find the nearest neuron (winner neuron) s_1 and the second nearest s_2 using squared Euclidean distance.
4. Increase the age of all the edges emanating from s_1.
5. Add the squared distance between the input signal and the winner neuron to a counter error of s_1 such as:

$$\Delta error(s_1) = \left\| w_{s_1} - \xi \right\|^2 \tag{1}$$

Move the winner neuron s_1 and its topological neighbours (neurons connected to s_1) towards ξ by a learning step ε_w and ε_n, respectively, of the total distance:

$$\Delta w_{s_1} = \varepsilon_w (\xi - w_{s_1}) \tag{2}$$

$$\Delta w_{s_n} = \varepsilon_n (\xi - w_{s_n}) \tag{3}$$

If s_1 and s_2 are connected by an edge, set the age of this edge to 0. If it does not exist, create it.
6. Remove the edges larger than a_{max}. If this results in isolated neurons (without emanating edges), remove them as well.
7. Every certain number λ of input signals generated, insert a new neuron as follows:
    - Determine the neuron q with the maximum accumulated error.
    - Insert a new neuron r between q and its further neighbour f:

$$w_r = 0.5(w_q + w_f) \tag{4}$$

Insert new edges connecting the neuron r with neurons q and f, removing the old edge between q and f.
    - Decrease the error variables of neurons q and f multiplying them with a constant α. Initialize the error variable of r with the new value of the error variable of q and f.
8. Decrease all error variables by multiplying them with a constant β.
9. If the stopping criterion is not yet achieved, go to step 2. (In our case the criterion is the number of neurons inserted)

Representation of 2D Objects with GNG

Given an image $I(x,y) \in \mathcal{R}$ we perform the transformation $y_T(x,y) = T(I(x,y))$ that associates to each one of the pixels its probability of belonging to the object, according to a property T. For instance, in figure 2, this transformation is a threshold function.

If we consider $\xi = (x, y)$ and $\mathcal{P}(x) = y_T(x)$, we can apply the learning algorithm of the GNG to the image I, so that the network adapts its topology to the object. This adaptive process is iterative, so the GNG represents the object during all the learning.

As a result of the GNG learning we obtain a graph, the Topology Preserving Graph $\mathcal{TPG} = \langle \mathcal{N}, C \rangle$, with a vertex (neurons) set $\mathcal{N}$ and an edge set C that connect them (figure 1). This $\mathcal{TPG}$ establishes a Delaunay triangulation induced by the object (O'Rourke, 2001).

Figure 2. Silhouette extraction
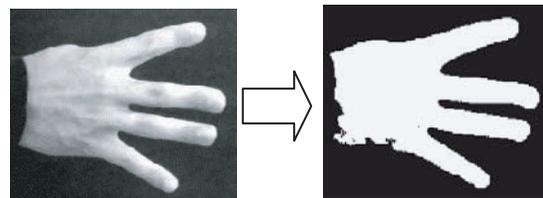
Representing Topology Deformations in Objects

The model is able also to characterize different parts of an object, or several present objects in the scene that had the same values for the visual property $\mathcal{T}$, without reset the different data structures for each one of the objects. This is due to the GNG capacity to divide itself into different parts when removing neurons and can be very useful to represent objects that change their topological structure breaking into small pieces or changing their shapes along a sequence of images. In this case a modification in the original algorithm of GNG must be done generating in step 2 a higher number of input signals to readapt from the previous map to the new image and avoiding steps 8 and 9 where neurons are deleted or added if necessary. None of the modifications of the original GNG algorithm to improve the robustness of the cluster analysis (Cselényi, 2005), (Cheng & Zell, 2000), (Qin & Suganthan, 2004), (Toshihiko, Iwasaki & Sato, 2003) use the structure of the map as a starting point to represent deformations in a sequence of objects shapes.

In this work GNG has been used to represent two-dimensional objects shape deformations in sequences of images, obtaining a topology representing graph.

When deformations in objects topology are small and gradual between consecutive frames in a sequence of images, we can use previous maps information to place the neurons without reset the learning process. Using this feature of GNG we achieve a high acceleration of the representation process.

For example in figure 3 are represented some objects with colour as a common feature in both images, that represent the same objects but as a foreground in white on the left and as a background in black on the right.

Experiments

To illustrate GNG capacities to represent topological deformations in objects, we have adapted the maps to an object shape that changes its topology from a compact square into four small squares in four steps (frames) obtaining graphs that represent the topology of the object shape along the images sequence but without reset the learning process for any image.

Figure 4 shows the original sequence of images used as input space for the self-organising map where from a homogenous square in the first image (on the left) four small squares are created in the last image (on the right). On the bottom of the figure are showed the results of the GNG adaptation establishing white

Figure 3. Representation of objects with similar visual properties as foreground and background
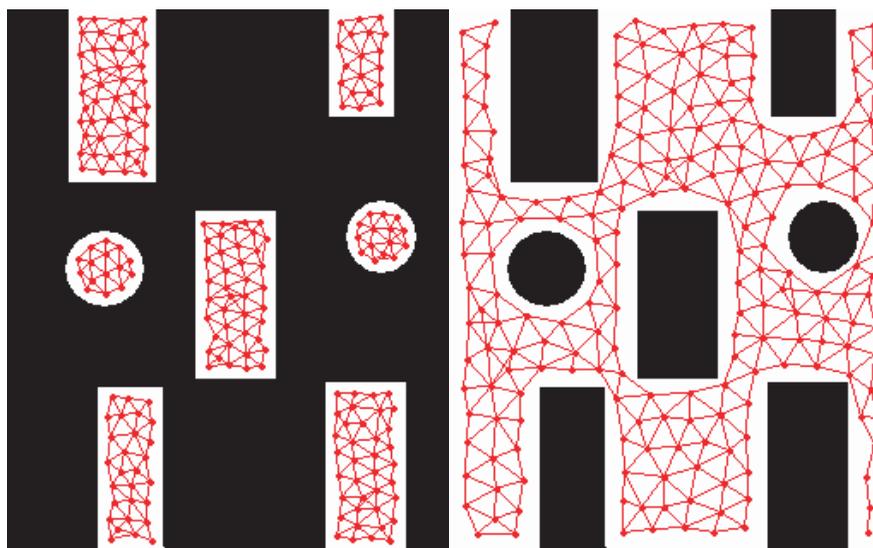
Figure 4. Results of GNG adaptation to changes in the input space
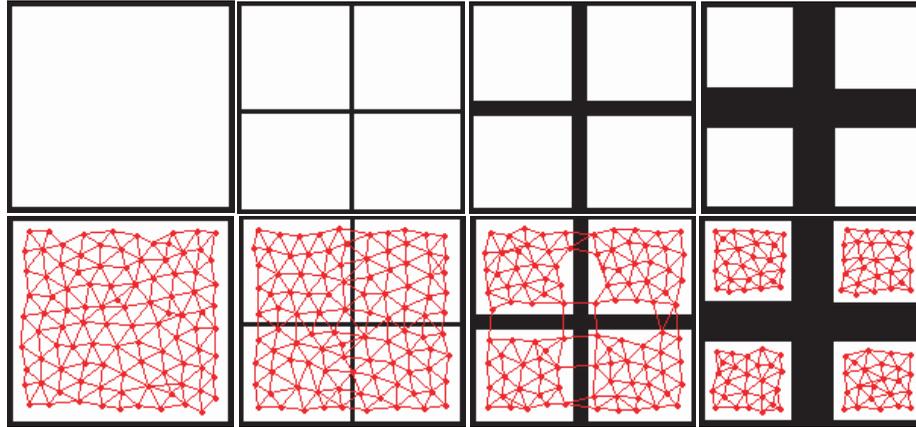
colour as a visual property of objects to be represented. From the first map (on the left), new maps are obtained based on the previous one without reset the learning process. This feature of GNG allows an acceleration of the images sequence representation.

As can be seen in the sequence of images, the map is able to separate the neurons into four groups representing the different squares in the original images when the distance between them is higher than the average of length of the edges that connects the neurons.

Figure 5 represents a sequence of deformations from a small circle to an ellipse and finally to a square used as input space to the GNG. The results of the adapta-

tion of the map without reset the learning algorithm between frames are showed.

The parameters used for the simulation are: N=100, $\lambda = 1000$ for the first map and 10000-20000 for the subsequent maps, $\varepsilon_w = 0.1$, $\varepsilon_n = 0.001$, $\alpha = 0.5$, $\beta = 0.95$, $\alpha_{max} = 250$.

The computational cost to represent a sequence of deformations is very low, compared with methods based on the adaptation of a new map for any frame of the sequence, since our method does not reset the algorithm for new frames. This feature provides the method with real-time capabilities.

Figure 5. Object deformation with GNG adaptation
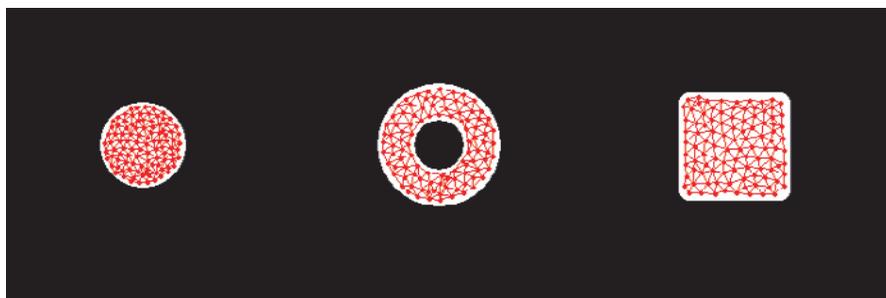
FUTURE TRENDS

The iterative and parallel performance of the presented representation model is the departure point for the development of high performance architectures that supply a characterization and tracking of non-rigid objects depending on the time available.

CONCLUSION

In this paper, we have demonstrated the GNG capacity of representation of bi-dimensional objects. Establishing a suitable transformation function, the model is able to adapt its topology to the shape of an object. Then, a simple, but very rich representation of the objects is obtained.

The model, by its own adaptation process, is able to divide itself so that it can characterize different fragments from an object or different objects in the same image. In addition, GNG can represent deformations in objects topology representing them along a sequence of images without reset the learning process. This feature accelerates the process of representation and tracking of objects.

REFERENCES

Flórez, F., García, J.M., García J. & Hernández, A. (2001). *Representation of 2D Objects with a Topology Preserving Network.* In Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems (PRIS'02), Alicante. ICEIS Press 267-276.

Flórez, F., García, J.M., García, J. & Hernández, A. (2001*). Hand Gesture Recognition Following the Dynamics of a Topology-Preserving Network.* I*n Proc. of the 5th IEEE Intern. Conference on Automatic Face and Gesture Recognition,* Washington, D.C. IEEE, Inc. 318-323.

Fritzke, B. (1995). *A Growing Neural Gas Network Learns Topologies.* In Advances in Neural Information Processing Systems 7, G. Tesauro, D.S. Touretzky T.K. Leen (eds.), MIT Press 625-632.

Kohonen, T. (2001). *Self-Organising Maps*. Springer-Verlag, Berlin Heidelberg.

Martinetz, T. & Schulten, K. (1994). *Topology Representing Networks*. Neural Networks, 7(3) 507-522.

O'Rourke, J. (2001). *Computational Geometry in C*. Cambridge University Press.

Ritter, H. & Schulten, K. (1986). Topology conserving mappings for learning motor tasks. In Neural Networks for Computing, AIP Conf. Proc.

Martinez, T., Ritter, H. & Schulten, K. (1990). *Three dimensional neural net for learning visuomotor coordination of a robot arm*. IEEE Transactions on Neural Networks, 1, 131-136.

Nasrabati, M. & Feng, T. (1988). *Vector quantisation of images based upon kohonen self-organising feature maps.* In Proc. IEEE Int. Conf. Neural Networks. 1101-1108.

Martinez, T. (1994). *Competitive hebbian learning rule forms perfectly topology preserving maps*. In ICANN.

Cselényi, Z. (2005). *Mapping the dimensionality, density and topology of data: The growing adaptive gas*. Computers Methods and Program in Biomedicine 78, 141-156.

Cheng, G. & Zell, A. (2000*). Double growing neural gas for disease diagnosis*. In Proceedings of ANNIMAB-1 Conference, 309-314.

Qin, A.K. & Suganthan, P.N. (2004). *Robust growing neural gas algorithm with application in cluster analysis.* Neural Networks 17 1135-1148.

Toshihiko, O., Iwasaki, K. & Sato, C. (2003). Topology representing network enables highly accurate classification of protein images taken by cryo electron-microscope without maskin. J*ournal of Structural Biology, 143,* 185-200.

KEY TERMS

Growing Neural Gas: A self-organizing neural model where the number of units is increased during the self-organization process using a competitive Hebbian learning for the topology generation.

Hebbian Learning: A time-dependent, local, highly interactive mechanism that increases synaptic efficacy as a function of pre- and post-synaptic activity.

Non-Rigid Objects: A class of objects that suffer deformations changing its appearence along the time.

Object Tracking: Is a task within the field of computer vision that consists on the extraction of the motion of an object from a sequence of images estimating its trajectory.

Self-Organising Neural Networks: A class of artificial neural networks that are able to self-organize themselves to recognize patterns automatically without previous training preserving neighbourhood relations.

Object Representation: Is the construction of a formal description of the object using features based on its shape, contour or specific region.

Topology Preserving Graph: Is a graph that represents and preserves the neighbourhood relations of an input space.

R

A Roadmap on Updates

Fernando Zacarías Flores
Benemérita Universidad Autónoma de Puebla, Mexico

Dionicio Zacarías Flores
Benemérita Universidad Autónoma de Puebla, Mexico

Rosalba Cuapa Canto
Benemérita Universidad Autónoma de Puebla, Mexico

Luis Miguel Guzmán Muñoz
Benemérita Universidad Autónoma de Puebla, Mexico

INTRODUCTION

Updates, is a central issue in relational databases and knowledge databases. In the last years, it has been well studied in the **non-monotonic reasoning** paradigm. Several semantics for logic program updates have been proposed (Brewka, Dix, & Knonolige 1997), (De Schreye, Hermenegildo, & Pereira, 1999) (Katsumo & Mendelzon, 1991). However, recently a set of proposals has been characterized to propose mechanisms of updates based on logic and logic programming. All these mechanisms are built on semantics based on **structural properties** (Eiter, Fink, Sabattini & Thompits, 2000) (Leite, 2002) (Banti, Alferes & Brogi, 2003) (Zacarias, 2005). Furthermore, all these semantic ones coincide in considering the AGM proposal as the standard model in the update theory, for their wealth in properties. The AGM approach, introduced in (Alchourron, Gardenfors & Makinson, 1985) is the dominating paradigm in the area, but in the context of monotonic logic. All these proposals analyze and reinterpret the AGM postulates under the **Answer Set Programming** (ASP) such as (Eiter, Fink, Sabattini & Thompits, 2000). However, the majority of the adapted AGM and update postulates are violated by update programs, as shown in (De Schreye, Hermenegildo, & Pereira, 1999).

UPDATES

Update theory deals with knowledge base represented by a propositional theory. Besides, deals with incorporating new knowledge about a dynamic world. This dynamism is due to knowledge comes from the real world, what means that knowledge evolves over time. This exchange rate mainly deals with changes in the extensional part of knowledge bases. However, the problem of updating the intensional part of a knowledge base (rules and descriptions of actions) remains basically unexplored. However, the problem of updates has attracted the researchers' attention in the last years who are dealing with such updates in the setting of logic programs. Though, some interesting proposals exist with foundation in Answer set programming (ASP), such as (Eiter, Fink, Sabattini & Thompits, 2000) (Leite, 2002) (Banti, Alferes & Brogi, 2003) (Osorio & Zacarias, 2003).

Answer set programming is a new paradigm used in the solution of the update issue. Particularly, this paradigm has taken bigger force around of update theory. A lot of theoretical work around updates under ASP has been developed by connoted researchers such as: Pereira, Alferes, Eiter, Osorio, Leite, Zacarias, and others. In the last years, a lot of theoretical work was devoted to explore the relationships between intuitionistic logic and ASP (Pearce, 1999) (Lifschitz, Pearce & Valverde, 2001). These results have recently provided a characterization of ASP by intuitionistic logic as follows: a literal is entailed by a program in the answer set semantics if and only if it belongs to every intuitionistically complete and consistent extension of the program formed by adding only negated literals (Pearce, 1999). The idea of these completions using in general intermediate logics is due to Pearce (Lifschitz, Pearce & Valverde, 2001). This logical approach provides the foundations to define the notion of non-monotonic inference of any propositional theory (using the standard connectives) in terms of a mono-

tonic logic (namely intuitionistic logic), see (Lifschitz, Pearce & Valverde, 2001) (Pearce, 1999).

STARTING WITH AGM

We start with an analysis on the AGM postulates and then we examine them with respect to update sequences. All these proposals are based on oneself principle of *causal rejection principle*. As is well known, if new knowledge of the world is somehow obtained, and it does not have conflicts with the previous knowledge then this new knowledge only expands knowledge. If by the contrary, new knowledge is inconsistent with the previous knowledge, and we want knowledge to be always consistent in all moment, we should solve this problem somehow. We point out that new information is incorporated into the current knowledge base subject to a *causal rejection principle*, which enforces that, in case of conflicts between rules, more recent rules are preferred and older rules are overridden.

An **update theory** is a knowledge base represented by a logic program. Then, let P be the program representing the current knowledge base, if it is updated by another program U, then P_U is a program updated of P if only if the models of P_U are the result of updating each of the models of P according to a given semantics S; to each of these models apply the update request U to obtain a new set of models *M*; P_U is any logic program whose models are exactly *M*.

The AGM approach proposes three basic operations on a belief set K: a) *expansion* K + Φ, which is simply adding the new information $\Phi \in \mathcal{L}_B$ to K. b) *revision* K * Φ, which is sensibly revising K in the light of Φ (in particular, when K contradicts Φ); and c) *contraction* K — Φ, which is removing Φ from K.

On the other hand, AGM proposes a set of postulates, K*1 — K*8, that any **revision operator** * mapping a belief set $K \subseteq \mathcal{L}_B$ and a sentence $\Phi \in \mathcal{L}_B$ into the revised belief set K * Φ should satisfy. We assume that K is represented by an epistemic state E, then the postulates K*1 — K*8 can be reformulated as in (Eiter, Fink, Sabattini & Thompits, 2000) as follows:

(K1) E * Φ represents a belief set.
(K2) $\Phi \in$ Bel(E * Φ).
(K3) Bel(E * Φ) $\subseteq$ Bel(E + Φ).
(K4) ¬Φ$\notin$Bel(E) implies Bel(E + Φ) $\subseteq$ Bel(E * Φ).
(K5) $\perp \in$Bel(E * Φ) only if Φ is unsatisfiable.

(K6) $\Phi_1 \equiv \Phi_2$ implies Bel(E * Φ_1) = Bel(E * Φ_2).
(K7) Bel(E * (Φ ^ γ)) $\subseteq$ Bel((E * Φ) + γ).
(K8) ¬γ $\notin$ Bel(E * Φ) implies Bel((E * Φ) + γ) $\subseteq$ Bel(E * (Φ ^ γ)).

Katsuno and Mendelzon (1991) proposed a set of postulates where a change Φ to a belief base B are propositional sentences over a finitary language. Some of the outstanding differences between the postulates of the AGM and those of Katsuno and Mendelzon are that revision should yield the same result as expansion E + Φ, providing Φ is compatible with E, which is not desirable for update in general. The postulate 8 says that if E can be decomposed into a disjunction of states (e.g., models), then each case can be updated separately and the overall result is formed by taking the disjunction of the emerging states.

Darwiche and Pearl (1997) have proposed postulates for iterated revision. This set of postulates is very simple and the majority of the adapted AGM and update postulates are violated by update programs. Another set of postulates for iterated revision, corresponding to a sequence E of observations, has been formulated by Lehmann (1995). Notice that in general the postulates proposed for iterated revision fail, and, with the exception of some postulates, each **change** is given by a single rule. Though, is that the two views described above amount to the same at a technical level.

All these approaches on the update issue consider it as a process of belief revision. However, following Gardenfors and Makinson (1991; 1994), **belief revision** can be related to non-monotonic reasoning by interpreting it as an abstract consequence relation on sentences, where the epistemic state is fixed. In the same way as Eiter we can interpret update programs as abstract consequence relation on logic programs. In spite of this, we should consider these proposals since for example Makinson (1993) considered a set of (desirable) **properties** for non-monotonic reasoning, and analyzed the behavior of some reasoning formalisms with respect to these properties.

Continuing with our research, immediately we comment in a general way the proposal of Alferes et. al., (2000). They introduced the concept of dynamic logic programs as a generalization of both the idea of updating interpretations through revision programs and of updating programs as defined by Alferes and Pereira (1997) and by Leite and Pereira (1997). Syntactically, dynamic logic programs are based on generalized logic

programs (GLPs), which allow default negation in the head of rules, but no strong negation whatsoever (Eiter, Fink, Sabattini & Thompits, 2000). The way in that the models of a update sequence are defined by Alferes et. al., is similar to the transformation used by Eiter et. al. These are defined as the **stable models** of the program resulting from a syntactic rewriting. This is called a dynamic update. Elements of the sequence are generalized logic programs.

Alferes et. al. defined in (Alferes & Pererira, 2002) its semantics by means of a dynamic logic programming generated by the sequence of commands. Afterwards, a translation of these commands, (a LUPS program) to a generalized logic program where stable models exactly correspond to the semantics of the original LUPS program. In this proposal the authors considering that the knowledge evolves from one knowledge state to another. Thus, given the current knowledge state KS, its successor knowledge state KS[U] is produced as a result of the occurrence of a non-empty set U of simultaneous updates. Each of the updates can be viewed as a set of actions and consecutive knowledge states are obtained as:

$$KSn = KS_0[U_1][U_2] \ldots [Un]$$

where U_i's represent consecutive sets of updates. This state is denote by:

$$KSn = U_1 \oplus U_2 \oplus \ldots \oplus Un$$

Thus, in dynamic logic programming the models of a sequence of updates are defined as the **stable models** of the program resulting from a syntactic rewriting. In (Alferes & Pererira, 2002) it is demonstrated that revision programs and dynamic updates are equivalent, provided that the original knowledge is extensional, i.e., the initial program contains only rules of the form A← or not A←.

One major difference can immediately be identified between our update programs and dynamic updates: In dynamic updates, the value of each atom is determined from the bottom level P_1 upwards towards Pn. the different evaluation strategy leads in effect to different semantics. Furthermore, Alferes et al. (2000) use a slightly non-standard concept of stable models. There is a semantic difference between dynamic updates and updates according to Eiter et.al (Eiter, Fink, Sabattini & Thompits, 2000).

On the other hand, one of the proposals more grateful on updates corresponds to (Eiter, Fink, Sabattini & Thompits, 2000). The authors in (Eiter, Fink, Sabattini & Thompits, 2000) redefine and implement an update process inspired in the proposal defined by Alferes et. al. you can refer to (Alferes & Pereira, 2002). The proposal (Eiter, Fink, Sabattini & Thompits, 2000) makes an exhaustive analysis of recent proposals based on non-monotonic logic. There, a syntactic redefinition of dynamic logic programs is presented, and semantically properties are investigated. In particular, a study on the dynamic logic programs verification of well known postulates of **belief revision** (Alchourron, Gardenfors & Makinson, 1985) is carried out. Also, structural properties of logic program updates are studied in (Eiter, Fink, Sabattini & Thompits, 2000). However, as happens in all works presented so far, most of the presented properties are not satisfied. This fact motivated our investigation to work towards a properties–based theory.

This is an approach to update non-monotonic knowledge bases represented as extended logic programs under the answer set semantics. They consider refinements of the semantics on the notion of minimality of **change**. This proposal proposes a mechanism for updates based on a sequence of logic programs. Informally, this program expresses layered derivability of a literal L, beginning from the top layer P_n and continuing downwards to the bottom layer P_1. The rule r layer Pi is only applicable if it is not refuted by a literal derived at a higher level that is compatible with H(r). Inertia rules propagate a locally derived value for L downwards to the first level, where the local value is made global.

Continuing in this direction, we have been working in finding properties that our **update operator** satisfies (Osorio & Zacarías, 2003) (Zacarías & Osorio, 2005) (Zacarías, Osorio, & Arrazola, 2005). Our purpose is to build a semantics based on structural properties. This is our main objective in the update theory. In (De Schreye, Hermenegildo, & Pereira, 1999) (Osorio & Zacarias, 2003) (Zacarías, Osorio & Arrazola, 2005) (Zacarias, 2005) the authors present a set of properties that the update operator satisfies. In this paper we continue with this same research line presenting a novel proposal with the aim to enrich the update theory that we have begun in (Osorio & Zacarias, 2003) (Zacarías, Osorio & Arrazola, 2005) (Zacarias, 2005). This novel proposal contributes with two benefits. First, we conserve many

of the properties presented in previous works (Osorio & Zacarias, 2003) (Zacarías, Osorio & Arrazola, 2005) (Zacarias, 2005), such as: **Weak Irrelevance of Syntax** (WIS). This property is similar to one postulate proposed by AGM, but in this case for **nonmonotonic logic** and under **Answer Set Programming** (ASP) introduced and defined by (Gelfond & Lifschitz, 1988).

On the other hand, we conclude that many approaches about program updates do not satisfy many of the properties defined in the literature (Alchourron, Gardenfors & Makinson, 1985) (Eiter, Fink, Sabattini & Thompits, 2000) (Katsuno & Mendelzon, 1991) (Banti, Alferes & Brogi, 2003). This is partly explained by the non-monotonicity of logic programs and the causal rejection principle embodied in the semantics, which strongly depends on the syntax of rules. Furthermore, we consider that a good update theory is based fundamentally on a set of properties.

As result of a first analysis of a proposal presented in (Eiter, Fink, Sabattini & Thompits, 2000), we introduced in (Osorio & Zacarías, 2003), a new update operator. This proposal satisfies several properties of AGM postulates, among them, a new property called *Weak Irrelevance of Syntax*. These properties give to an agent an added value with respect to other proposals that do not satisfy them. It is necessary to highlight the simplicity of our proposal, which allows to an agent to be able to respond in a correct and opportune way.

Continuing our analysis on updates we present our main results about updates of logic programs: a properties-based approach published in (Zacarias, 2005). In this proposal we presented several properties on theory updates. We consider these properties from a non-monotonic reasoning perspective, by naturally interpreting program updates as non-monotonic consequence relations. In this proposal we consider our properties under N logic. Additionally, we have presented in (Zacarías, 2005) some examples about updates on answer set programming.

In (Zacarías, 2005) we have introduced a new proposal towards the enrichment of the update operator "⊕". There, we have presented a refinement of the stable model semantics for the update operator. Also, we presented a new property that allows us to face updates where new information contains rules that define a **conservative extension**. So, we gave an extension of our properties proven in (Osorio & Zacarías, 2003), under N logic. This approach is based on the work made by

Eiter et al. (Eiter, Fink, Sabattini & Thompits, 2000), and inspired in a recent approach presented by Alferes et al. (Banti, Alferes & Brogi, 2003). With this work, we improve and enrich the **update operator** proposed by Eiter et al. (Eiter, Fink, Sabattini & Thompits, 2000), giving as result a new update operator.

FUTURE TRENDS

Just as in (Eiter, Fink, Sabattini & Thompits, 2000) we coincide that because of apparent lack of minimality of change, we then considered refinements of the semantics in terms of minimal and strictly minimal answer sets. Several issues remain for further work. An interesting point (Eiter, Fink, Sabattini & Thompits, 2000) concerns the formulation of **postulates** (**principles or properties**) for update operator on logic programs and, more generally, on non-monotonic theories. As you can see in (Eiter, Fink, Sabattini & Thompits, 2000), several postulates from the area of logical **theory change** fail for update programs. This may be explained by the dominant role of syntax for update embodied by causal rejection of rules.

CONCLUSIONS

In this paper, we considered a new proposal to provide an **update process** to our agents. Our proposal is a novel and simple methodology that allows an agent to maintain updated its knowledge base in all moment. This provides an agent to behave in a rational way, similar to human behavior. Furthermore, it is an appropriate proposal for applications that require answers in real time. Also, this proposal opens the possibilities for building real-life applications, like intelligent agents whose rational component is modelled by a knowledge base, which is in turn maintained using update logic programs.

REFERENCES

Alchourron C.E., Gardenfors P., & Makinson D. On the logic of Theory Change, Partial Meet Functions for Contraction and Revision Functions. Journal of Symbolic Logic, 50:510-530, 1985.

Alferes J.J. & Pereira L.M. Logic programming updating –a guided approach- in computational Logic: Logic programming and Beyond, Essays in honour to Robert A. Kowalski, Part II. Springer Verlag, 382-412, 2002

Ariely O., Denecker M., Van Nuffelen & Bruynooghe M.. Database repair by Signed formulae In D. Seipel and J.M. Turrul, editors, Foundations of Information and Knowledge Systems, Third International symposium, FoIKS 2004, Wilhelminenburg Castle, Austria, vol. 2942 LNCS, pp. 231–241, 2004.

Banti F., Alferes J. & Brogi A. A principled semantics for logic program updates. In M. Gelfond, N. Leone and P. Pfeifer, editors, proceedings into Eighteenth International Join Conference, LNAI, México, Springer Verlag, 2003.

Brewka G., Dix J., & Knonolige K. Nonmonotonic reasoning, an overview. CSLI Publication Eds. Leland Stanford Junior University, 1997.

De Schreye D., Hermenegildo M. & Pereira L.M. Paving the Roadmaps: Enabling and Integration Technologics, 1999.

Eiter T., Fink M., Sabattini G., & Thompits H. Considerations on Updates of Logic Programs. In M.O. Aciego, LP. de Guzmßn, G. Brewka, and L.M. Pereira, editors, Proc. Seventh European Workshop on Logic in Artificial Intelligence JELIA 2000, vol. 1919 in Lecture Notes in Artificial Intelligence. LNAI, Springer 2000.

Gelfond M., & Lifschitz V. The stable model semantics for logic programs. Proceedings of the Fifth International Conference on Logic Programming 2, MIT Press. Cambridge, Ma. pp.1070-1080, 1988.

http://www.compulog.org/net/Forum/Supportdocs.html

Katsumo H. & Mendelzon A.O. On the difference between updating a knowledge base and revising it. in: J.A. Allen, R. Fikes and E. Sandewell. eds.. Principles of knowledge representation and reasoning: Proceedings of the Second International Conference (Morgan Kautmann. San Mateo. CA. 1991) pp. 387-394.

Leite J.A. Evolving Knowledge Bases – Specification and Semantics. PhD thesis, Departamento de Inform´atica, Universidade Nova de Lisboa, 2829-526, 2002.

Lifschitz V., Pearce D., & Valverde A. Strongly equivalent logic programs. ACM Transactions on Computational Logic, 2:526-541, 2001.

Osorio M. & Zacarias F., Irrelevance of Syntax in updating answer set programs, Proceedings Of Fourth Mexican International Conference On Computer Science Enc'03, pp.183-188, Eds. J. H. Sossa, and E. Perez, México, 2003.

Pearce D. From Here to There: Stable negation in Logic Programming, in D. Gabbay, H. Wansing (Eds.) What is Negation? Kluwer Academic Publishers, Dordrecht.

Zacarías F., Osorio M., & Arrazola J. Updates based on Structural Properties –USP-. Gests international transactions on computer science and engineering, pp. 61-72, issn: 1738-6438, isbn: 89-953729-5-8, October 2005.

Zacarías F. Belief Revision and Updates in Commonsense Reasoning, Ph. D thesis, Universidad de las Américas Puebla, 2005.

Zacarías F. & Téllez A. Programación lógico–funcional. In CONIELECOMP 2002, pages 45–49, Acapulco (México), 2002.

KEY TERMS

Beliefs: An agent whose knowledge base is the theory *T* believes *F* if and only if *F* belongs to every intuitionistically complete and consistent extension of *T* by adding only negated literals.

Causal Rejection Principle: Which enforces that, in case of conflicts between rules, more recent rules are preferred and older rules are overridden.

Equivalence: Two programs are *equivalent* if they have exactly the same answer sets.

Expansion: Which is simply adding the new information A to knowledge base KB.

Principle of Irrelevance of Syntax: The meaning of the knowledge that results from an update must be independent of the syntax of the original knowledge, as well as independent of the syntax of the update itself.

Update: Let P be the program representing the current knowledge base, if it is updated by another program U, then P_U is a program updated of P if only if the models of P_U are the result of updating each of the models of P according to a given semantics S; to each of these models apply the update request U to obtain a new set of models *M*; P_U is any logic program whose models are exactly *M*.

Weak Irrelevance of Syntax: T_1 T_2 implies *Bel(K* $T_1) = Bel(K$ $T_2)$, where *K*, T_1 and T_2 are any theories, Bel(*T*) defines the set of answer sets of *T*, is the update operator, and understanding that equivalence means that both programs (T_1 and T_2) have the same answer sets.

A Robot Model of Dynamic Appraisal and Response[1]

Carlos Herrera
Intelligent Systems Research Centre, University of Ulster, Northern Ireland

Tom Ziemke
University of Skovde, Sweden

Thomas M. McGinnity
Intelligent Systems Research Centre, University of Ulster, Northern Ireland

INTRODUCTION

A general goal of biologically inspired robotics is to learn lessons from actual biological systems and to find applications in robot design. Neural controllers and adaptive algorithms are major tools to model, at some level of abstraction, functions, structures, and behaviors present in biological systems. This involves, of course, identifying in virtue of what biological systems exhibit the behavioral characteristics we want to explore. One of the biological phenomena of great interest is *emotion*. Despite the effort of leading researchers to raise the question "whether machines can be intelligent without any emotions" (Minsky., 1988), AI interest in emotional phenomena has increased only in the last decade. An underlying assumption is that many cognitive functions, such as memory, attention, learning, decision making and planning, are at least partly based on emotional mechanisms in biological systems (Damasio, 1995).

One of the qualities of emotional behavior is its flexibility (Frijda, 1986), which contrasts with the rigidity of stereotyped behaviors such as reflexes or habits. Hence, it is relevant to investigate what it is that makes emotional behavior flexible. The body, through mostly chemical channels, produces diffuse effects on the neural system, processes at the root of emotional phenomena. Parisi has recently argued that in order "to understand the behavior of organisms more adequately we also need to reproduce in robots the inside of the body of organisms and to study the interactions of the robot's control system with what is inside the body"

(Parisi, 2004), using the term *internal robotics* to denote the study of the interactions between the (neural) control system and the rest of the body.

Mechanisms that control homeostasis, based on hormonal modulation, can motivate appropriate behaviors (Avila-García & Cañamero, 2004; Gadanho & Hallam, 2001). Emergent behaviors from the interaction of a motivational system with the environment may be called emotional. Cañamero's architecture, for example, consists of "a set of motivations; a repertoire of behaviors that can satisfy those internal needs or motivations as their execution carries a modification in the levels of specific variables; and a set of 'basic' emotions."(Cañamero, 2005).

We consider emotional phenomena to emerge from a dynamic interaction between internal states, current perceptions and environmental relations, such that certain neural/physiological states have a close causal link with relational situations. This is, in a nutshell, the *embodied appraisal* hypothesis (Carlos Herrera, 2002; Prinz, 2004). We use two major concepts from the dynamical systems (DS) approach to cognition (Clark, 1997; Kelso, 1995): *collective variables* and *control parameters*. In (Carlos Herrera, 2002) we argue that internal states can be interpreted as collective variables of agent/ environment interaction that allow tracing concern-relevant situations. These variables are "non-specific: they do not prescribe or contain a code for the emerging structure" (Kelso, 1995). They also can be considered control parameters, as activation in the agent's physiological substrate affects overall

action readiness (response, including perceptual and cognitive readiness).

BACKGROUND

An architecture for the design of emotional appraisal and response in artificial agents must take into account that emotions bear an intrinsic dynamic relationship between internal mechanisms, embodiment and situation (Frijda, 1993; Lazarus, 1991; Lewis, 2005). Emotions are emergent patterns that involve relational behavior as well physiological and psychological processes. In this section we argue that physiological states are essential for understanding emotion appraisal and response: they allow to trace agent-environment relations, and their modification is a mechanism for control of dynamics.

Appraisal is the process by which an agent is capable of recognizing that a situation is relevant to some of its concerns. From an information-processing perspective, an agent requires the capacity to differentiate situations which anticipate that a concern may be at stake if no proper response is carried out. Cognitive models consider appraisal the product of a reasoning engine (Zajonc, 1980), and robotic models often simplify this problem by manipulating the environment so that the concern-relevance of specific objects/stimuli is particularly salient (e.g. red color for dangerous objects).

Appraisal involves categorization, or hot cognition (Zajonc, 1980). The theory of embodied appraisal argues that the body plays an essential role in structuring sensory-motor patterns that, once processed by the brain, result in appraisal (Damasio, 2000). In the case of emotion certain physiological states are indicative of concern-relevant situations (Prinz, 2004). A high level of adrenaline, for instance, correlates with a wide class of emotional situations. The fact that the correlation is not one-to-one (physiological states are not sufficient to determine emotions) does not imply that they have no relationship to interactive relations. We understand embodied appraisal as dynamical coupling (attunement) in which some internal states are representative (collective variables) of agent/environmental interactions.

But emotion is not only about appraisal, but also response. Emotion theorists have proposed the notion of *action tendency* to explain the inherent relational purpose of emotional behavior: it establishes or modifies a relationship between the agent and the world "at large" (Frijda, 1986). That means, "[a]ction tendencies are hypothesized … for theoretical reasons: to account for latent readiness and to account for behavior flexibility" (Frijda, 1986). Tendencies imply a direction, although they are "not usually guided by a prior goal representation" (Frijda, 1986). It is also important to distinguish between action tendency and the function of emotional behavior. For example, the tendency in fear is withdrawal. The function, on the other hand, is protection. Similarly, the tendency in shock or surprise is interruption of ongoing activity, whilst the function is reorientation (Frijda, 1986). Withdrawal can come as, for example, freeze, flight, or faint; responses with very different functional roles. Hence, emotions are far from reflex-like responses. Even though emotion responses are often stereotyped and the product of evolution, we "should not conceive affect programs as fixed and peremptory" (Lazarus, 1991), i.e. "[t]o the extent that action programs are fixed and rigid, action tendency loses much of its meaning" (Frijda, 1986). On the contrary, emotional responses are dynamically situated, that is, outward behavior is configured in dynamic interaction with the environment.

For modeling the mechanisms underlying action tendencies in biological agents, it should be noted how different physiological subsystems are dynamically interrelated. In particular, certain hormones (e.g. adrenaline), can affect, on the one hand, the autonomic system, whose activation involves a process of energy mobilization, and thus action readiness (Frijda, 1986). Hormones also act as neuromodulators, affecting the general processing of the nervous system, thus producing forms of cognitive or attention readiness.

MAIN FOCUS OF THE CHAPTER

This article presents a robotic approach for the emergence of a coupled agent-environment interaction with the ability to: (a) to appraise the concern-relevance of the situation, and (b) to control through activation of action readiness. The specific mechanisms that allow the emergence of such phenomena are based on sensitivity to overall patterns of interaction through the production of hormonal regulation.

The model illustrated in Figure 1 (Carlos. Herrera, 2006) is intended to illustrate the relationships between nervous system, body and world. The basic feature is that a number of internal variables in the body (such as

Figure 1. Left: Model of dynamic appraisal in embodied interaction. Relation between (a) physiological states and situations. (b,c) nervous system and rest of the body, (e, d) world and body. Right: Proposed controller for a Khepera robot (prey).
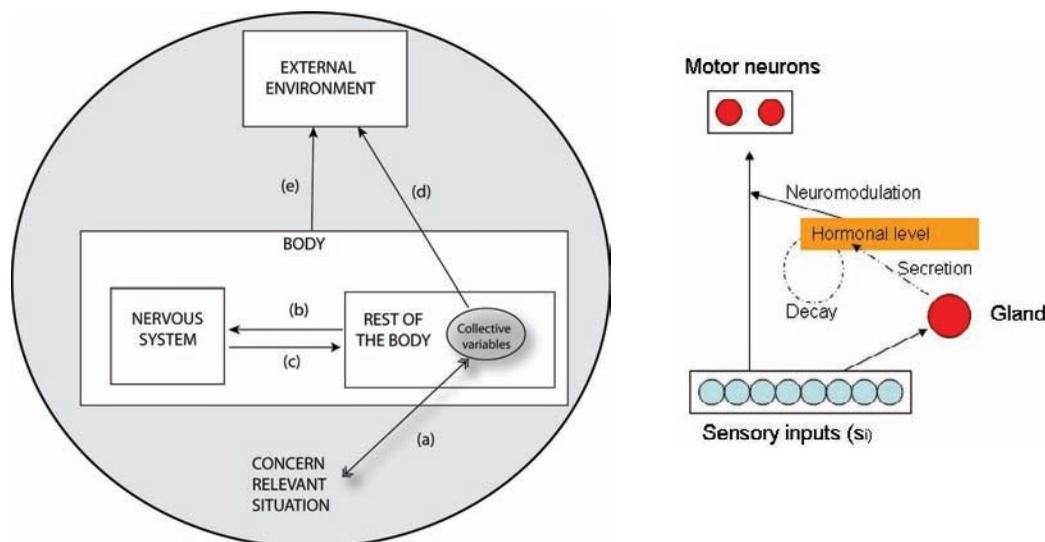
hormonal levels) allow the agent to trace the dynamics of some concern-relevant aspects of the relationship to the environment. Their processing can be conceived a simple form of body maps or feeling (Damasio, 1995, 2000).

This activation of internal states is integrated with current cognitive, perceptual and sensory-motor processes. The nervous system participates in the homeostatic balance, as hormonal production is a function of nervous activation – thus the collective variables are to some extent control variables: a change produces a change in action readiness, reflected in the dynamic relationship towards concern-relevant aspects of the situation. Sensory-motor activity (relationship of the nervous system to the environment), in conjunction with further nervous processing (secondary appraisal), produces a change in action tendency. Emotional behavior is the result of this process.

Experimental Setup

In this section we present a simple, preliminary experimental setup to the implementation of this model (Carlos. Herrera, 2006). We apply an evolutionary robotics approach evolving connection weights in

the neurocontrollers of simulated Khepera robots in a predator/prey scenario inspired by (Nolfi & Floreano, 1998). Two robots, equipped with infrared sensors, are placed in random positions in a square environment surrounded by walls. The predator, which also has a camera, is rewarded for hitting the prey, which in turn is rewarded for avoiding the predator. Both robots are controlled by feedforward networks that map sensory inputs and to motor outputs/activations. To allow rich relational dynamics, the maximum speed of the prey is set to twice the predator's.

We abstractly model some of the functions of the endocrine system as a simple gland that secretes one type of hormone. The resulting hormonal level is intended to be a collective variable of the interaction, i.e. produce a function that allows us to trace concern-relevant situations. The model also requires that the hormone level has an effect on the generation of behavior through modulation of the neural controller. In order to achieve this, we feed it to the neural controller as an extra input. We have established the level of hormonal secretion as a function of the activity of the sensory cells and a fixed rate of hormonal absorption.

As mentioned above, the level of hormonal release is intended to be a collective variable of the interaction.

In order to achieve this, we paid attention to what situations are concern-relevant in the prey/predator scenario. We are interested in situations in which there is danger of being caught. Despite several possible strategies of approach and avoidance, given the speed of the prey, danger is most present when the prey is caught between predator and walls, whereas if no walls interfere, the prey can produce optimal escape behaviors. Extrapolating this observation, if a robot is near a wall and a predator, the sum of the activation of all sensors will be larger than when only one of them is present. Therefore it makes sense to establish a linear relationship between sensory activation and hormonal release, in line with intensity theories of emotion that relate emotion elicitation to "densities of neural firing" (Tomkins, 1962) (we do not claim intensity theories to be complete, though). We therefore define hormonal release as the sum of the activation of all other sensors. The decay function of the hormone level, or rate of absorption, is set to 2, i.e. at every time step, the hormone level will be divided by 2. The resulting level is fed back in to the neural controller as an extra input, playing the role of a parametric bias with a neuromodulatory effect on the motor output. The hormonal level can thus be expressed as a function of current level and sensory states as follows:

$$E_t = \left(E_{t-1} + \sum_{i=1.8} S_i \right) / 2$$

The controller weights of the controller and the modulation effect of the hormonal level are evolved, while the production and absorption of hormones is kept fixed. If, as assumed, the level of hormone is significant of a class of situations (danger), then the robot can be expected to use it for the evolution of adaptive emotional behavior.

Results

As with other experiments in co-evolution, performance of prey and predator along the evolutionary process are co-dependent, and cycles can be observed. It is nevertheless possible to analyze cross generational strategies and fitness (Nolfi & Floreano, 1998). Due to limited space, we will here only analyze the behavior of a single generation (100). In this analysis we will verify whether: (a) the hormonal level can be considered a collective variable of the dynamics of interaction, that is, it allows us to track situations in which the prey is between wall and predator, (b) the hormonal level acts as a parametric bias for the neural controller so as to generate an action tendency that changes the relationship to the environment (whose function is to safeguard the prey's concern, i.e. to escape), and (c) the resulting behavior shows a degree of flexibility measured as robustness in unforeseen circumstances.

Figure 2 shows the prey's behavior and its relationship to the hormone level. A high hormonal level modulates the normal behavior (circular), producing a straightforward fast motion (right). This change in behavior is correlated to dangerous situations (caught between predator and wall). The prey thus is capable of appraising concern-relevant situations, by means of attunement through an appropriate collective variable

Figure 2. Left: Interaction between predator (black, discontinuous trace) and prey. At the marked points (1, 2, 3, 4) the prey was caught between wall and predator. Center: level of hormonal activation throughout the interaction. Right: close up of an escape behavior.
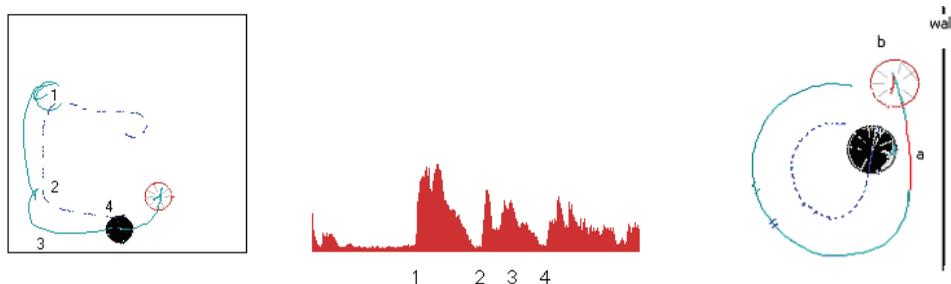
Table 1. Performance in original and modified environments

	No Obstacles	One Obstacle	Two obstacles
Prey escapes	56%	40%	33%
Predator kills	39%	39%	36%
Prey crashes	5%	21%	31%

represented by a simulated physiological function. Further analysis of this behavior and control dynamics can be found in (Carlos Herrera, Ziemke, & Moffat, 2006).

In terms of flexibility our hypothesis was that, in any scenario in which the concern-relevance of the robot is represented by such a variable, the prey's ability for escape should not be seriously reduced. We tested the performance of prey and predator with one and two obstacles placed in random positions, for 1000 runs (note that the original evolutionary process was carried out without obstacles). Given that the evolved predator cannot determine whether a prey is hiding behind an obstacle, we exclude the runs in which the predator crashed without entering the prey's sensor range.

As the table shows, environmental changes do affect the prey's ability to escape. The presence of obstacles has a limited effect on the functionality of the prey's behavioral strategy for escaping the predator, although the proportion of times where the prey crashes increases significantly. Comparing the flexibility with that of evolved reactive controllers, we found that the rate of successful escapes decreases more rapidly in favor of predator catches, while crashes remain stable (Carlos. Herrera, 2006). We could draw a parallel from such results to real emotional behavior: if we react emotionally to dangerous situations, we will be more likely to escape, but also more likely to harm ourselves. Being involved in a fast fleeing behavior implies that the danger of bumping into further danger increases. This is congruent with emotion theory "not all behavior elicited by emotional events can be considered coping activity … Instrumental behavior, too, shows dysfunctional or non-functional features, among them sheer disturbance manifestations like decreased precision of skilled movements" (Frijda, 1986).

FUTURE TRENDS

Despite increasing interest in the modeling of emotions in robotics, it remains one of the cornerstones of Artificial Intelligence. In this article, we have presented a dynamical-embodied approach. Some of the obvious limitations of this initial experimental setup should be avoided in further work. For instance, it would be interesting to let the control of the 'endocrine gland', as well as the modulation of behavior co-evolve or co-develop, and to investigate the role evolution and learning may play in emotional attunement. This would involve research into the capacities of neural networks to learn temporal patterns of concern-relevance. We have here also considered the physiological system only in its relationship to the nervous system, and not in its relation to body dynamics, leaving aside the autonomic system and its energy mobilization role. This allowed us to functionally replace the physiological system by a one-dimensional hormone level. More complex robotic physiologies, exploring relationships between body states and their relationship to sensors, motors, and nervous system should be investigated.

There are two short term experimental goals for further work. First, given that it is not always possible for the designer to identify concern-relevant situations and possible ways to trace them through internal mechanisms, and in search of increased robotic autonomy, we plan to find self-organizing techniques to achieve similar results. In particular, we are considering the use of evolution and learning, as well as novel mechanisms such as anticipatory networks and liquid state machines, to allow an internal structure to identify and gain sensitivity to such situations. A more realistic model of the physiological systems involved is also necessary, as arguably the control architecture presented here is 'just' a form of recurrent neural network. Finally, we will explore the role of a hormonal regulation system

within a framework of behavioral attractors, in order to be able to carry out detail dynamical system analysis of parametric/behavioral biases and the resulting action tendencies.

CONCLUSION

In this paper we suggest a biologically inspired approach to flexible behavior through emotion modeling. We consider emotion to emerge from relational interaction of body, nervous system and world, through sensory-motor attunement of internal parameters to concern-relevant relationships. We interpret such relationships with the notions of collective variable and control parameters.

The preliminary experiments presented here indicate the viability of the model and the potential benefits for robotic behavioral flexibility. If interaction between an agent and the environment can be conceived as a dynamical system with certain collective variables, it is worthwhile considering the possibility for the agent to embody emotional mechanisms that allow the agent to trace such collective variables.

The approach presented involves an alternative way of considering the problem of adaptation and cognition from a dynamical system perspective.

- Collective variables and control parameters are considered useful for the construction of adaptive dynamical systems.
- Perception is not directed exclusively to objective features of the environment, but first and foremost to establishing agent/environment relationships and the attunement of action tendencies.
- Third, neural/computational processing is not the only mechanism for extracting such global characteristics, but physiological aspects of embodiment, e.g. hormonal modulation, can carry out essential information-processing functions too.

In summary, this approach suggests that instead of considering sensory stimuli as the primary basis for perception, proprioceptive feedback plays a constitutive role in gaining appropriate information about agent/environment relations, and thus the environment itself. This approach can enlighten the notion of hot cogni-tion and its relation to standard cognitive/perceptual mechanisms and representational content.

R

REFERENCES

Avila-García, O., & Cañamero, L. (2004). Using Hormonal Feedback to Modulate Action Selection in a Competitive Scenario. *From Animals to Animats, 8*, 243–252.

Cañamero, L. (2005). Emotion understanding from the perspective of autonomous robots research. *Neural Networks, 18*(4), 445-455.

Clark, A. (1997). *Being There: Putting Brain, Body, and World Together Again*: MIT.

Damasio, A. (1995). *Descartes' error: emotion, reason, and the human brain*: Picador.

Damasio, A. (2000). *The feeling of what happens: body, emotion and the making of consciousness*: Vintage.

Frijda, N. (1986). *The Emotions*: Editions de la Maison des sciences de l'homme.

Frijda, N. (1993). The place of appraisal in emotion. *Cognition and emotion, 7*(3-4), 357-387.

Gadanho, S., & Hallam, J. (2001). Robot Learning Driven by Emotions. *Adaptive Behavior, 9*(1), 42.

Herrera, C. (2002). *Emotions And Perception: On The Role Of Prioceptive Feedback.* Paper presented at the IASTED 2002. Special session on Perception and Emotions, Malaga.

Herrera, C. (2006). *The synthesis of emotion in artificial agents.*, Glasgow Caledonian University, Glasgow.

Herrera, C., Ziemke, T., & Moffat, D. (2006). *Emotions as a bridge to the environment: the role of body in organisms and robots.* Paper presented at the The Ninth International Conference on the SIMULATION OF ADAPTIVE BEHAVIOR (SAB'06).

Kelso, J. (1995). *Dynamic Patterns: Self-organization of Brain and Behavior*: Mit Pr.

Lazarus, R. (1991). *Emotion and Adaptation*: Oxford University Press US.

Lewis, M. (2005). Bridging emotion theory and neurobiology through dynamic systems modeling. *Behavioral and Brain Sciences, 28*(02), 169-194.

Minsky., M. (1988). *The Society of Mind*: Touchstone.

Nolfi, S., & Floreano, D. (1998). Co-evolving predator and prey robots: Do 'arm races' arise in artificial evolution? *Artificial Life, 4*(4), 311-335.

Parisi, D. (2004). Internal robotics. *Connection Science, 16*(4), 325-338.

Prinz, J. (2004). Embodied Emotions. In *Thinking about Feeling: Contemporary Philosophers on the Emotions* (pp. 44-59).

Tomkins, S. (1962). *Affect, imagery, consciousness*. New York: Springer.

Zajonc, R. (1980). Feeling and thinking: Preferences need no inferences. *American Psychologist, 35*(2), 151-175.

KEY TERMS

Action Readiness / Tendencies: Physiological states affect the readiness for engagement in certain dynamics of the interaction

Artificial Emotion: The attempt to synthesize in robots or artificial systems some of the functional properties of emotion.

Embodied Appraisal: Theory that asserts sensitivity to concern-relevant situations is facilitated by physiological and homeostatic mechanisms in an embodied agent.

Emotions: Phenomena present in biological systems by which an adaptive agent is capable of appraising the concern-relevance of situations and provide flexible responses through generation of physiological, cognitive and behavioral readiness.

Collective Variable / Control Parameter: In dynamical systems theory, collective variables allow tracing global dynamic patterns, control parameters lead the system through such patterns.

Concerns: The conditions under which a system can continue to function.

Hormonal Modulation: Change in the functionality of neural, sensory and motor systems achieved through changes in hormonal levels.

Neuro-Robotics: Approach to robot control through the use of neural networks.

ENDNOTE

[1] This paper represents a revised version of the paper "Behavioral flexibility: An emotion based approach", presented at the IWANN'07 conference. This work has been partly supported by a European Commission grant to the project *"Integrating Cognition, Emotion and Autonomy"* (IST-027819,) as part of the European *Cognitive Systems* initiative.

Robots in Education

Muhammad Ali Yousuf
Tecnologico de Monterrey – Santa Fe Campus, Mexico

INTRODUCTION

The new paradigm in engineering education demands hands-on training of the students using technology oriented projects. The roots of this approach can be traced back to the work of **Seymour Papert** in 1970s when he built a programmable turtle with a reflective light sensor (Papert, 1971). His ideas ultimately lead to the educational theory of **constructionism** (Papert, 1986 and Harel & Papert, 1991). According to this theory, students learn very effectively when they are involved in the creation of an external object that lives in the real world. Learners use this object to think with, and to relate ideas of, their subject of inquiry (Bourgoin, 1990). From an educational point of view, the theory of Papert can be linked to the **constructivist** theory of Jean Piaget (Paiget, 1972). According to this theory, learning comes from an active process of knowledge construction. This knowledge can be gained through real life experiences and linked to a learners' previous knowledge. The concept of turtle was evolved further at MIT and became the famous **Programmable Brick** by **Fred Martin** who also developed new learning environments and methodologies based on this concept (Martin, 1988 and Martin 1994). The unusual idea put forward by the Brick, at least at the time of its invention, was the incorporation of the "design" work into the learning process. Students were not only users in this case, but were actively involved in the design process, while solving their problems (Martin, 1996a). The 'Brick' was later adopted and incorporated by the LEGO MINDSTORMS kit (RCX in 1998 and NXT made available in 2006). The use of the name "MINDSTROMS" can also be traced back to the book by Seymour Papert (Papert 1980). Versions of these Bricks for economically challenged communities have also been proposed recently (Sipitakiat, et al, 2004).

The **active learning** methodology (Harmin and Toth, 2006) uses this philosophy of involving students in their own learning through class discussions and group problem solving and proves to be effective at least in certain cases. Robots have become a major player in this area and have been employed in improving the quality and level of student learning, ranging from primary schools to graduate level. As pointed out by Resnick and Martin (Resnick and Martin, 1990), "Creatures built from Electronic Bricks fall on the fuzzy boundary between animals and machines, forcing students to come to terms with how machines can be like animals, and vice versa". In engineering courses incorporating connectionism approach, the students are asked to design and program a robot for a specific task. They also work in small teams and help and learn from each other.

However it is important to know what is currently available to an educator so that he/she can develop the required skills, abilities, attitudes and values in students. In this article we identify some of the major research centres working in the area of education utilizing robots and discuss some of the robotic kits now available to educators. We also comment on the famous robotic competitions worldwide.

BACKGROUND

Many researches have tried to include a project-oriented approach to the teaching of engineering subjects. This approach has the benefit of allowing students to seek information on their own while developing a well defined product. The use of robots in enhancing the quality of education at a university level has been discussed by many authors (Takahashi et al, 2006, Gage & Murphy 2003, Matsushita et al 2006). Students from school to undergraduate level have been involved in microcontroller based robotic projects. They can design, build and test their robots themselves and that helps them later in their education. Mukai and McGregor (Mukai and McGregor, 2004) have gone to the level of teaching control to eight graders in public schools.

Robots can help educators in teaching and learners in learning various branches of basic sciences. This is in addition to their obvious use in engineering courses. Mathematics (Algebra, geometry, matrices, calculus),

Physics (electricity, force, Newton's laws, momentum, rotations and angular momentum) are few examples (Yousuf et al., 2006). Their connection to biology comes through understanding and linking of human sensors to robotic or electronic sensors. Bratzel (Bratzel, 2005) uses engineering principles to teach physics and physical science by incorporating LEGO robots. She introduces, in chapters of increasing difficulty, concepts of motion, forces, fluids, stability, work and energy, etc. Bratzel has also correlated the activities in her book with the national science content (in USA) standards for grades 5-12, and hence makes for a good choice for educators at that level.

The purpose of integrating robotics is not just to create excitement among students but to use this excitement to help them in learning what they find difficult to learn using conventional methods. All the educators want to develop certain abilities, values and attitudes among their students. Some of the international accrediting organizations, like the Accreditation Board for Engineering and Technology recommend the use of a "**competency-based learning**" methodology for course development (Earnest, 2005 and Criteria for Accrediting Engineering Programs, 2006). The core of this system is that all activity (in the classroom, laboratory or projects) must be focused on pre stated competencies by using structured learning objectives. This system also demands the evaluation to be based on the competencies developed by the students. This can only be done by looking at concrete evidence (e.g., electrical or mechanical systems developed, software, technical reports, etc). Once again, the use of robots provides the educator with well defined competencies to be evaluated precisely.

EDUCATIONAL ROBOTS

We divide this part into subsections discussing various aspects of the main theme. Since each of these subjects is sufficiently broad in itself, we concentrate on a few representative cases only.

Research Groups in the Area of Educational Robotics

This is an area of immense activity and the number of research groups active in this area is extremely large. Almost all the robotic research groups have an interest in the educational aspects of the subject. Many have tried to involve their own undergraduate students into the process and have gained new and deeper insights into student behaviour and learning.

Massachusetts Institute of Technology

MIT has been active in the area of robotics for a long time but it was in 1989 that Fred Martin (Martin 2001) started a worldwide movement in educational robotics by introducing his now famous undergraduate design contest. This was also the launching of the corresponding robot "brain" called the Handyboard (Handyboard). It is now being used by educators worldwide together with the Interactive C language to program the system (Butler et. al., 2006). This system is powerful enough to have industrial applications. The work of Fred Martin was continued later with Mitchel Resnick's Life-Long Kindergarten group (Kindergarten Group). This work was partly sponsored by the LEGO Group and became the foundation for the LEGO MINDSTORMS Robotics Invention System (to be discussed later).

NASA Robotics Alliance Project

The Robotics Alliance Project is an initiative of the **NASA**, the National Aeronautics and Space Administration in USA. It is based on the idea that NASA is going to need many more robot engineers for its space endeavours in the future and the only way to have quality engineers in the future is to invest in their training (RAP). Hence the project starts at the level of K-12 and does it through a variety of robotics programs, competitions and curriculum development. Their web site offers links to curriculum resources starting from primary to doctoral level. It also lists some of the major robotic competitions, and internship opportunities for students, etc. NASA also provides video and webcast archives for educators.

Carnegie-Mellon University

The robotics institute at the **Carnegie-Mellon University** is one of the largest of its type and has various projects with an educational impact. The CREATE project, which is an acronym for Community Robotics, Education and Technology Empowerment (CREATE) has research programs in curriculum design for teaching robot programming at the secondary school level and

beyond. They are also developing curriculum that will help middle and high school educators. Another valuable contribution by the same group is the development of a fully accredited robotic exploration course for high school juniors. The course is offered in summer and allows students to build robots using special fast-build kits. These kits have also been designed at CMU and include even a vision system allowing students to develop rover missions in the classroom and home environments (Nourbakhsh et al., 2005). Students then go home together with the robot. This way student can keep working on the subject after leaving the center.

Fraunhofer Institut Intelligente Analyse und Informationssyteme

The **Fraunhofer AIS** in Germany (Fraunhofer), sponsored by the Federal Ministry of Education and Research, is active in the educational aspects of robotics. They have developed a robot called Roberta, which conveys the knowledge about engineering and computer science to youngsters in an exciting way. Their particular focus is female population. Dozens of tutors have been guided in the use of this methodology and a few hundred students (more than three-fourth girls) have been trained. Tutors get training at the Fraunhofer AIS with specially developed teaching material to support learning. A national network of regional Roberta centers is being established to support tutors locally, to ensure nation-wide exchange of experience, and to disseminate the results of this project.

Educational Robot Kits

Educational robotic kits provide the users everything needed to build and program a robot. Some of them are more flexible than others, but each comes with its own programming language or programming environment. We discuss here some of them.

LEGO MINDSTORMS RCX and NXT Robots

As mentioned in the introduction above, it was an idea developed at MIT and introduced to the mass market by the LEGO Group (LEGO) in 1998. The **LEGO MINDSTORMS** robots are perhaps the simplest kits to start with, yet they are general and broad enough to be used as pedagogic platforms for training even at the

university level. Banking upon the students' familiarity with LEGO (those who are not familiar need very little extra time to start with), and utilizing a specially developed, highly visual programming language, the system helps kids from six years upward to learn and enjoy robotics. The system comes complete with online tutorials and is backed by innumerable web sites, books and tutorials. The newer version is called MINDSTORMS NXT and is even more flexible and powerful with some new sensors added and some of the older sensors upgraded. The new servo motors have been fitted with rotation sensors, allowing precise position control.

Parallax Systems

Parallax, Inc., (Parallax) is a developer of electronic systems (including robots) generally for higher level students though they do have systems for ages eight and above. Scribbler™ Robot for example, is meant for first-time programmers and roboticists age eight and up. The more advanced systems designed for experienced users include the Toddler® Robot, QuadCrawler and HexCrawler robots. The big advantage for educators is the large number of books, manuals and curriculum material available for these systems. Most of it is available free from the company web site and / or included in the kits.

Fishcertechnik Reconfigurable Robot Kits

The **Fischertechnik** systems (Fischerwerke Artur Fischer Gmbh) developed in Germany, are some of the most advanced robotic invention systems available. These systems allow students of all ages (even adults) to enjoy the field with flexible robotic kits. They also provide curriculum material for those willing to incorporate Fischertechnik into their classes. These systems are easy to program and come with a variety of sensors, motors, LEDs, etc. The kits can be used to teach advanced concepts in engineering too, including PLCs or Programmable Logic Control.

Robotic Competitions

Robotic competitions are an ideal way to keep the interest of the students alive and to give them a well-defined target to achieve. Most of the competitions also give very strict guidelines as to what can be used in

the construction of the system and who can participate. Many of the international competitions discourage teacher participation in the final presentation and hence allow students to grow and develop into self responsible persons. These competitions also grade students based on their group work, cleanliness, and presentation skills. In a nutshell, they are an excellent way of "standardizing" curriculum and assessment.

According to the Manchester based organization, "For Inspiration and Recognition of Science and Technology (**FIRST**), the FIRST Robotics Competition involves around 32,500 students in 2007. The junior version, called the FIRST LEGO League has been designed for children in age group 9-14 years. An estimated number of 88,000 children participated in this activity in 2006. There are dozens of other local robotic competitions all over the world for which there is no statistics available. However a search on any of the Internet search engines brings a large number of pages (e.g. the Robot Competition FAQ). Many of these are confined to a university, college or school. But in many cases the models can be followed and replicated at other places. The famous 6.270 Autonomous Robot Design Competition at MIT (MIT 6.270) is a good example and has been running successfully for more than two decades. Most of the research groups mentioned above may also be contacted as they frequently arrange national level competitions.

FUTURE TRENDS

The field of educational robots is full of promising directions. One important factor is the development of generic robot systems and the standardization of corresponding robot control software. **Microsoft Robotics Studio** (Microsoft Robotics Studio) is one of the most recent efforts in the direction of software standardization. It allows ANY robot to be controlled through a single platform. Companies like Parallax have already developed free examples for users to try on their boebot. On the hardware side also we are going to see more modular systems with flexibility and extensibility. The **MIT Tower** (Lyon, 2003) is a typical product of this type. Though not yet commercially available, it allows the user to start with a basic system and then to go on adding functionality depending upon requirements. Currently available modules are for sensing, actuation, data storage, and infrared communication. They plan to add new ones for enhanced display output and high-speed wireless communication, etc.

CONCLUSION

In this article we have made an attempt to provide a brief overview of the state of the art in educational robotics, the work of major research groups and the offerings of commercial vendors for educators. Comments on the future directions in educational robotics have been made. Most of the efforts in this direction have started to take advantage of the experiences of others and we hope to see more balanced and well though-out curricula to be developed for each of the major areas of basic sciences.

REFERENCES

Bratzel, B., (2005). Physics by Design, College House Enterprises, LLC.

Bourgoin, M. O. (1990). Children using LEGO robots to explore dynamics. In Harel, I. (Ed.), Constructionist Learning. MIT Media Laboratory, Cambridge, MA.

Butler, D., Strohecker, C., and Martin, F. (2006). Sustaining Local Identity, Control and Ownership While Integrating Technology into School Learning. Book chapter in *Lecture Notes in Computer Science*, Volume 4226/2006, Springer Berlin / Heidelberg.

CREATE Lab at Carnegie Mellon University, Robotics Institute 5000 Forbes Avenue, Pittsburgh, PA 15213, http://www.ri.cmu.edu/

Criteria for Accrediting Engineering Programs - Effective for Evaluations During the 2006-2007 Accreditation Cycle. Published by the Accreditation Board for Engineering and Technology (ABET), 111 Market Pl., Suite 1050, Baltimore, MD 21202, http://www.abet.org/

Earnest, J. (2005), ABET engineering technology criteria and competency based engineering education, Frontiers in Education, FIE '05. Proceedings 35th Annual Conference, 19-22 Oct. 2005 Page(s): F2D - 7-12.

FIRST, 200 Bedford St., Manchester, NH 03101, http://www.usfirst.org

Fischerwerke Artur Fischer Gmbh & Co. KG, Weinhalde 14 - 18, 72178 Waldachtal Deutschland, http://www.fischertechnik.com/

Fraunhofer-Gesellschaft zur Förderung, der angewandten Forschung e.V., Postfach 20 07 33, 80007 München, Germany, http://www.fraunhofer.de/

Gage, A., Murphy, R.R., (2003) Principles and experiences in using legos to teach behavioral robotics, Frontiers in Education, FIE 2003, 33rd Annual, Volume 2.

Handyboard, The; An authorized supplier is Gleason Research, P.O. Box 1494, Concord MA 01742 web site, http://handyboard.com/

Harel, I. and Papert, S. (eds) (1991). Situating Constructionism, Constructionism, Norwood, NJ: Ablex Publishing.

Harmin, M. and Toth, M, (2006) Inspiring Active Learning: A Complete Handbook for Today's Teachers, Association for Supervision & Curriculum Deve; 2nd Edition.

Kindergarten Group at the MIT Media Lab, Building E15 77 Massachusetts Avenue, Cambridge, MA 02139-4307 USA, http://llk.media.mit.edu/

LEGO MINDSTORMS, LEGO Company, Global Company Communications, DK-7190 Billund, Denmark, http://mindstorms.lego.com

Lyon, C. (2003). *Encouraging Innovation by Engineering the Learning Curve*, Cambridge, MA: Department of Electrical Engineering and Computer Science Master's Thesis, Massachusetts Institute of Technology.

Martin, F. G. (1988). *Children, cybernetics, and programmable turtles*, Master's thesis,

The Massachusetts Institute of Technology, MIT Media Laboratory, 20 Ames Street Room E15–315, Cambridge, MA 02139.

Martin, F. G. (1994). *Circuits to Control: Learning Engineering by Designing LEGO Robots*, Ph.D. thesis, Massachusetts Institute of Technology, MIT Media Laboratory, 20 Ames Street Room E15–315, Cambridge, MA 02139.

Martin, F. (1996a). *Kids Learning Engineering Science Using LEGO and the Programmable Brick.* Presented at the annual meeting of the American Educational Research Association, April 8–12, 1996, New York, NY.

Martin, F. (1996b). Ideal and real systems: A study of notions of control in undergraduates who design robots, in *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World* (Yasmin Kafai and Mitchel Resnick, eds.), Lawrence Erlbaum

Martin, F. (2001). *Robotic Explorations: A Hands-on Introduction to Engineering,* Prentice Hall.

Matsushita, K., Yokoi, H., Arai, T., (2006), Robotics in Education: Plastic Bottle Based Robots for Understanding Morph-Functionality, The 15th IEEE International Symposium on Robot and Human Interactive Communication, ROMAN 2006.

Microsoft Robotics Studio, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-7329, USA, http://www.microsoft.com/

MIT 6.270, Autonomous Robot Design Competition, http://web.mit.edu/6.270/

Mukai, H. and McGregor, N. (2004). Robot control instruction for eighth graders, *Control Systems Magazine*, IEEE, Volume 24, Issue 5, Page(s):20 – 23.

Nourbakhsh, I., Crowley, K., Bhave, A., Hamner, E., Hsiu, T., Perez-Bergquist, A., Richards, S., and Wilkinson, K., (2005). The Robot Autonomy Mobile Robotics Course: Robot Design, Curriculum Design and Educational Assessment, *Autonomous Robotics Journal*, 18(1).

Papert, S. (1971). *Teaching children thinking*, Cambridge, MA: MIT Artificial Laboratory Memo no. 247, Massachusetts Institute of Technology.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, New York.

Papert, S. (1986). *Constructionism: A new opportunity for elementary science education*, proposal to the National Science Foundation. MIT Media Laboratory.

Parallax, Inc, 599 Menlo Drive, Rocklin, California 95765, USA, http://www.parallax.com/

Piaget, J. (1972). *The Principles of Genetic Epistemology*, N. Y.: Basic Books.

RAP Robotics Alliance Project at NASA, NASA Headquarters, Suite 5K39, Washington, DC, 20546-0001, USA, http://www.nasa.gov/index.php.

Resnick, M and Martin, F (1990). Children and Artificial Life, *E&L Memo* No. 10, MIT Media Laboratory, Cambridge, Massachusetts.

Robot Competition FAQ, http://robots.net/rcfaq.html

Sipitakiat, A., Blikstein, P., and Cavallo, D., (2004). GoGo Board: Augmenting ProgrammableBricks for Economically Challenged Audiences, In *Proceedings from International Conference of the Learning Sciences*, California, USA, June, pp. 481-488, 2004.

Takahashi, Y., Uchiyama, Y., Takagi, H., Takashima, T.(2006), University Robotics Education with Fabrication Experiences of Twelve-Axis Biped Robot, SICE-ICASE, 2006. International Joint Conference.

Yousuf, M. A., De la Cueva, V. and Montúfar, R. (2006). Learning Two-Dimensional

Physics and Mathematics through their Applications in Robotic Manipulators, *International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 06)*. University of Bridgeport, USA.

KEY TERMS

Active Learning: The methodology which demands students to participate actively in their own learning, guided and supervised by the educator.

Competency Based Learning: A system of learning which represents a dynamic mixture of knowledge, understanding, capacity and ability. The competencies are measurable outcomes of learning and hence can be evaluated at the end of the process.

Constructionsim: According to the constructionist learning theory, people learn most effectively when they are involved in the creation of an external artefact in the world. This artefact becomes an "object to think with," which is used by the learner to explore and embody ideas related to the topic of inquiry (Martin, 1996b).

Constructivism: An educational theory or school of learning, based on the idea that knowledge is constructed by the learner based on mental activity. Learners create a mental image of how the world operates and they adapt and transform their understanding using their earlier knowledge.

Industrial Robotic Manipulators: Mechanical arms used in industry, with sensor feedback and automatic control software.

Mobile Robots: Robots with the capability to move autonomously from one place to the other, including wheeled, legged, submerged and flying robots, etc.

Pedagogy: The art (or science) of being a teacher but commonly referred to as the technique used in instruction.

Robust Learning Algorithm with LTS Error Function

Andrzej Rusiecki
Wroclaw University of Technology, Poland

INTRODUCTION

Feedforward neural networks (FFNs) are often considered as universal tools and find their applications in areas such as function approximation, pattern recognition, or signal and image processing. One of the main advantages of using FFNs is that they usually do not require, in the learning process, exact mathematical knowledge about input-output dependencies. In other words, they may be regarded as model-free approximators (Hornik, 1989). They learn by minimizing some kind of an error function to fit training data as close as possible. Such learning scheme doesn't take into account a quality of the training data, so its performance depends strongly on the fact whether the assumption, that the data are reliable and trustable, is hold. This is why when the data are corrupted by the large noise, or when outliers and gross errors appear, the network builds a model that can be very inaccurate.

In most real-world cases the assumption that errors are normal and iid, simply doesn't hold. The data obtained from the environment are very often affected by noise of unknown form or outliers, suspected to be gross errors. The quantity of outliers in routine data ranges from 1 to 10% (Hampel, 1986). They usually appear in data sets during obtaining the information and pre-processing them when, for instance, measurement errors, long-tailed noise, or results of human mistakes may occur.

Intuitively we can define an outlier as an observation that significantly deviates from the bulk of data. Nevertheless, this definition doesn't help in classifying an outlier as a gross error or a meaningful and important observation. To deal with the problem of outliers a separate branch of statistics, called robust statistics (Hampel, 1986, Huber, 1981), was developed. Robust statistical methods are designed to act well when the true underlying model deviates from the assumed parametric model. Ideally, they should be efficient and reliable for the observations that are very close to the assumed model and simultaneously for the observations containing larger deviations and outliers.

The other way is to detect and remove outliers before the beginning of the model building process. Such methods are more universal but they do not take into account the specific type of modeling philosophy (e.g. modeling by the FFNs). In this article we propose new robust FFNs learning algorithm based on the least trimmed squares estimator.

BACKGROUND

The most popular FFNs learning scheme makes use of the backpropagation (BP) strategy and a minimization of the mean squared error (mse). Until now, a couple various robust BP learning algorithms have been proposed. Generally, they take advantage of the idea of robust estimators. This approach was adopted to the neural networks learning algorithms by replacing the mse with a loss error function of such a shape that the impact of outliers may be, in certain conditions, reduced or even removed.

Chen and Jain (1994) proposed the Hampel's hyperbolic tangent as a new error criterion, with the scale estimator β that defines the interval supposed to contain only clean data, depending on the assumed quantity of outliers or current errors values. This idea was combined with the annealing concept by Chunag and Su (2000). They applied the annealing scheme to decrease the value of β, whereas Liano (1996) introduced the logistic error function derived from the assumption of the errors generated with the Cauchy distribution. In a recent work Pernia-Espinoza et al. (2005) presented an error function based on tau-estimates. An approach based on the adaptive learning rate was also proposed (Rusiecki, 2006). Such modifications may significantly improve the network performance for corrupted training sets. However, even these approaches suffer from several difficulties and cannot be considered as universal (also

because of properties of applied estimators). Besides, very few of them have been proposed until today and they exploit the same basic idea, so we still need to look for new solutions.

ROBUST LTS LEARNING ALGORITHM

Least Trimmed Squares

The least trimmed squares estimator (LTS), introduced by Rousseuw (1984, 1985) is a classical high break-down point robust estimator, similar to the slower converging least median of squares (LMS) (Rousseuw, 1984). The estimator and its evaluations are often used in linear and nonlinear regression problems, in sensitivity analysis, small-sample corrections, or in simple detecting outliers. The main difference between the LTS estimator and the least sum of squares, but also M-estimators, is obviously the operation performed on residuals. In this case however, robustness is achieved not by replacing the square by another function but by superseding the summation sign with something else. The nonlinear least trimmed squares estimator is then defined as:

$$\hat{\theta} = \arg\min_{\theta \in R^p} \sum_{i=1}^{h} (r^2)_{i:n} \tag{1}$$

where $(r2)1{:}n \leq \ldots \leq (r2)n{:}n$ are the ordered squared residuals $ri2(\theta) = \{yi - \eta(xi, \theta)\}2$, y_i represents the dependent variable, $x_i = (x_{i1}, \ldots, x_{ik})$ the independent input vector, and $\theta \in R^p$ denotes the underlying parameter vector for the general nonlinear regression model. The trimming constant h must be chosen as $n/2 < h \leq n$ to provide that $n-h$ observations with the largest residuals do not directly affect the estimator. Under certain assumptions the estimator should be robust not only to outliers (Stromberg, 1992) but also to the leverage points (grossly aberrant values of x_i) (Rousseuw, 1987).

Derivation of the LTS Algorithm

For simplicity, let us consider a simple three layer feedforward neural network with one hidden layer. The net is trained on a set of n training pairs:

$\{(x_1, t_1), (x_2, t_2), \ldots, (x_n, t_n)\}$,

where $x_i \in R^p$ and $t_i \in R^q$. For the given input vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$, the output of the jth neuron of the hidden layer may be obtained as:

$$z_{ij} = f_1\left(\sum_{k=1}^{p} w_{jk} x_{ik} - b_j\right) = f_1(inp_{ij}) \quad \text{, for } j=1,2,\ldots,l, \tag{2}$$

where $f1(\cdot)$ is the activation function of the hidden layer, wjk is the weight between the kth net input and jth neuron, and bj is the bias of the jth neuron. Then the output vector of the network $yi = (yi1, yi2, \ldots, yiq)T$ is given as:

$$y_{iv} = f_2\left(\sum_{j=1}^{l} w'_{vj} z_{ij} - b'_v\right) = f_2(inp_{iv}) \quad \text{, for } v=1,2,\ldots,q. \tag{3}$$

Here $f_2(\cdot)$ denotes the activation function, w'_{vj} is the weight between the vth neuron of the output layer and the jth neuron of the hidden layer, and b'_v is the bias of the vth neuron of the output layer. Now, we introduce the robust LTS error criterion, based on the Least Trimmed Squares estimator. The new error function is defined as:

$$E_{LTS} = \sum_{i=1}^{h} (r^2)_{:n} . \tag{4}$$

In this case, $(r^2)_{1:n} \leq \ldots \leq (r^2)_{n:n}$ are ordered squared residuals of the form

$$r_i^2 = \left\{\sum_{v=1}^{q} |(y_{iv} - t_{iv})|\right\}^2 . \tag{5}$$

The trimming constant h must be carefully chosen because it is responsible for the quantity of patterns suspected to be outliers.

We assume, for simplicity, that weights are updated according to the gradient-descent learning algorithm but this can be extended to any other gradient-based algorithm. Then to each weight is added (α denotes a learning coefficient):

$$\Delta w_{jk} = -\alpha \, \partial E_{LTS} / \partial w_{jk}, \tag{6}$$

R

$$\Delta w'_{vj} = -\alpha\ \partial E_{LTS}/\partial w'_{vj}, \qquad (7)$$

where

$$\partial r_i/\partial w_{jk} = f'_2(inp_{iv})w'_{vj}f'_1(inp_{ij})x_{ik}, \qquad (8)$$

and

$$\partial r_i/\partial w'_{vj} = f'_2(inp_{iv})z_{ij}. \qquad (9)$$

The main problem that may occur here is calculating the E_{LTS} derivative. It is not continuous and it can be written as:

$$\frac{\partial \sum_{i=1}^{h}\left(r^2\right)_{i:n}}{\partial r_i} = \begin{cases} 2r_i\ for & r_i^2 \leq \left(r^2\right)_{:h} \\ 0\ for & r_i^2 > \left(r^2\right)_{:h} \end{cases} \qquad (10)$$

As it was experimentally demonstrated, such shape of the derivative function is smooth enough for the BP learning algorithm.

In the use of robust learning algorithms, there exist some problems, concerning mainly the choice of a starting point for the method. In fact, we can divide it into two tasks: choosing initial network parameters, and choosing the right scale estimator. If the initial weights of the network are not properly selected, the learning process may move in the wrong direction and the algorithm may stack in a local minimum. In this case the network performance might become very poor. The scale estimator or its equivalent (here, the trimming constant h) is responsible for the amount of outliers that are to be rejected during the training, it's clearly evident then, that if h is incorrect, gross errors may be regarded as good data and desired points may be discriminated.

Following (Chen and Jain, 1994), we decided to use our LTS robust algorithm after a period of training by the traditional BP algorithm to set the initial parameters. We proposed two strategies of choosing the trimming parameter h. In the first approach we assumed a predefined value of h, depending on expected percentage of outliers in the training data (LTS1). In this case, additional a-priori knowledge of the error distribution is needed, so the strategy is not very useful. The second approach (LTS2) is to choose h by using the median of all errors as:

$$h = |\{r_i : |r_i| < c * median(|r_i|),\ i=1\ldots n\}|, \qquad (11)$$

where $c=1.483$ for the MAD scale estimate (Huber, 1981). Errors used for calculating h were the errors obtained after the last epoch of the traditional back-propagation algorithm, so the value of h is set constant for the training process.

Simulation Results

The LTS learning algorithm was tested on function approximation tasks. In this paper we present only a few of many different testing situations. The first function to be approximated is $y=x^{-2/3}$ proposed by Chen and Jain (1994), the second one is a two-dimensional spiral given as $x=\sin y$, $z=\cos y$.

To simulate real data containing noise and outliers we used different models, defined as follows:

- Clean data without noise and outliers;
- Data corrupted with the Gross Error Model: $F=(1-\delta)G+\delta H$, where F is the error distribution, $G\sim N(0.0,0.1)$ and $H\sim N(0.0,10.0)$ are Gaussin noise and outliers and occur with probability $1-\delta$ and δ (data Type 1);
- Data with high value random outliers (Type 2), proposed in (Pernia-Espinoza et al., 2005) of the form $F=(1-\delta)G+\delta(H_1+H_2+H_3+H_4)$, where $H_1\sim N(15,2)$, $H_2\sim N(-20,3)$, $H_3\sim N(30,1.5)$, $H_4\sim N(-12,4)$.
- Data with outliers generated from the Gross Error Model, injected into the input vector x_i (Type 3).

The performances of the traditional backpropagation algorithm (BP), robust LMLS algorithm, and the both variations of the novel robust LTS algorithm, LTS1 and LTS2, were compared.

Looking at the Table 1 we can see that for the clean data of the first task, all algorithms act relatively well. For the data containing gross errors, the two variations of the LTS present the best performance and it is hard to say, which of them is better, while for the data with high value outliers only LTS2 and LMLS ensure good fitting to the testing data, while LTS1, though still better than the BP algorithm, acts rather poor.

For the data containing outliers injected into input, the algorithms LTS1 and LTS2 presented the best per-

Table 1. The mean MSE for the 100 trials for the networks trained to approximate function of one variable

	Clean Data	Data with gross errors (Type 1)		Data with high value outliers (Type 2)		Data with gross errors in the input vector (Type 3)	
Algorithm	δ=0.0	δ=0.1	δ=0.2	δ=0.1	δ=0.2	δ=0.1	δ=0.2
BP	0.0007	0.0398	0.0809	1.7929	4.0996	0.0140	0.0180
LMLS	0.0007	0.0061	0.0088	0.0050	0.0053	0.0151	0.0177
LTS1	-	0.0054	0.0056	0.0632	0.1454	0.0104	0.0120
LTS2	0.0013	0.0049	0.0067	0.0051	0.0061	0.0112	0.0149

Table 2. The mean MSE for the 100 trials for the networks trained to approximate two-dimensional spiral

	Clean Data	Data with gross errors (Type 1)		Data with high value outliers (Type 2)	Data with gross errors in the input vector (Type 3))	
Algorithm	δ=0.0	δ=0.1	δ=0.2	δ=0.1	δ=0.1	δ=0.2
BP	0.0000	0.3967	0.7722	24.9154	0.0014	0.0057
LMLS	0.0000	0.0584	0.1442	0.0682	0.0006	0.0034
LTS1	-	0.0318	0.0390	1.7108	0.0001	0.0023
LTS2	0.0006	0.0284	0.0534	0.0311	0.0007	0.0023

Figure 1. Simulation results for the network trained to approximate one dimensional function (data Type 1): backpropagation algorithm (dash- dot line), LMLS alg. (dashed line), LTS1 alg. (dotted line), LTS2 alg. (solid line)

R
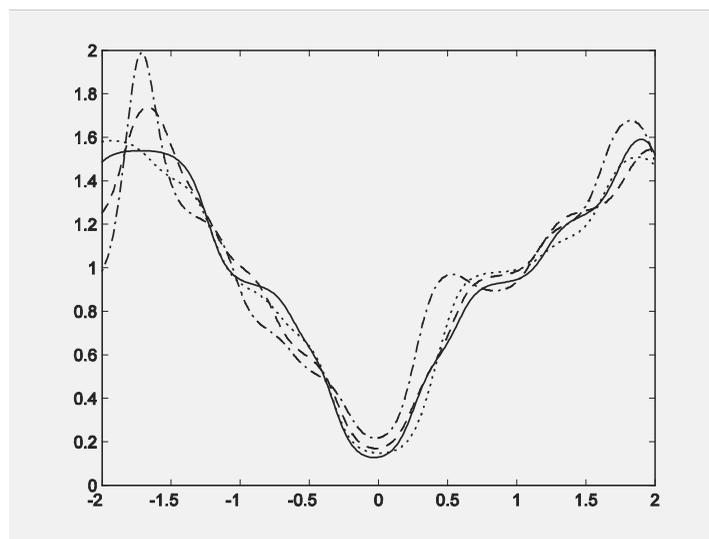
Figure 2. Simulation results for the network trained to approximate one dimensional function (data Type 3): backpropagation algorithm (dash- dot line), LMLS alg. (dashed line), LTS1 alg. (dotted line), LTS2 alg. (solid line)
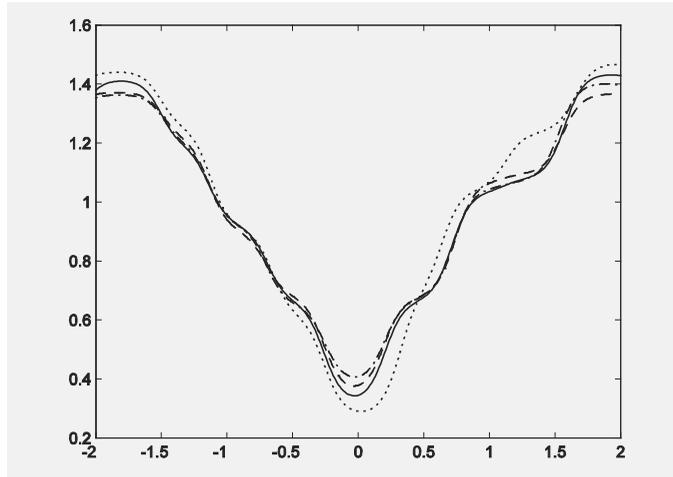
Figure 3. Simulation results for the network trained to approximate two-dimensional spiral (data Type 1): backpropagation algorithm (dash- dot line), LMLS alg. (dashed line), LTS1 alg. (dotted line), LTS2 alg. (solid line)
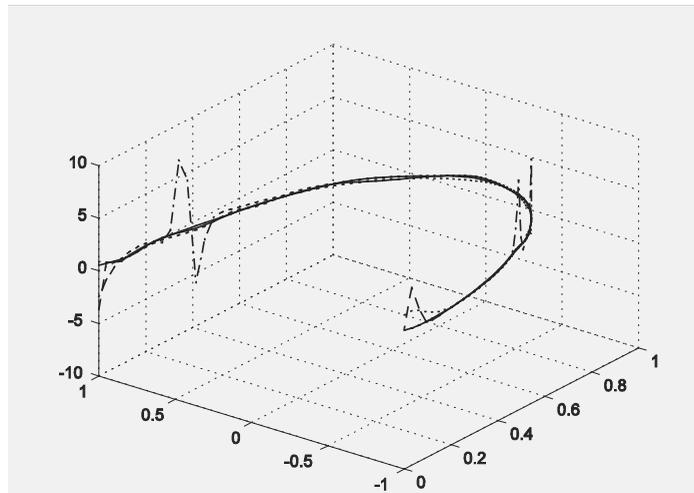
formance and the error of the LTS1 is even over 25% better than for the Lmls and BP.

Results obtained for the second approximation task are, generally, similar. For the data containing outliers, the superiority of the LTS algorithm is clearly evident. The LTS2 acts well also for the high value outliers, showing the lowest error. Besides, for gross errors in the input vector also the LTS1 and LTS2 appear to be the best.

To summarize, one can notice that both LTS algorithms showed performance better than other two algorithms, for the data containing gross errors in the input, as well as in the output vector.

FUTURE TRENDS

Potentially, robust learning algorithms, based on modified error function, can be designed also for training other types of NN structures, such as recurrent or self-organizing networks. Moreover, for the FFNs there is plenty of techniques (adaptive learning rate, transfer functions, etc.) that can be used to make their learning process more robust to outliers.

CONCLUSION

In this paper a novel robust LTS learning algorithm was proposed. As it was experimentally demonstrated, it behaves better than traditional algorithm, and robust Lmls algorithm, in the presence of outliers in the training data. Moreover, it is simultaneously the first robust learning algorithm that takes into account also gross errors injected into the input vector of the training patterns (leverage points). Especially in its second version (LTS2), with median error used to set the trimming constant h, it can be considered as simple and effective mean to increase learning performance on the contaminated data sets. It doesn't need any additional a-priori knowledge of the assumed error distribution to ensure relatively good training results in any conditions. The robust LTS learning algorithm can be easily adapted to many types of neural networks learning strategies.

REFERENCES

Chen, D.S., Jain, R.C. (1994). A robust back propagation learning algorithm for function approximation. IEEE Transactions on Neural Networks, vol. 5, pp. 467-479

Chuang, C., Su, S., Hsiao C. (2000). The Annealing Robust Backpropagation (ARBP) Learning Algorithm. IEEE Transactions on Neural Networks, vol. 11, pp.1067-1076

Hagan, M. T., Demuth, H. B., Beale, M. H. (1996). Neural Network Design. Boston, MA: PWS Publishing

Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A. (1986). Robust Statistics the Approach Based on Influence Functions. John Wiley & Sons, New York

Hornik, K., Stinchconbe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, vol. 2, pp. 359-366

Huber, P.J. (1981). Robust Statistics. Wiley, New York, 1981

Jacobs, R.A. (1988). Increased rates of convergence through learning rate adaptation. Neural Networks, vol. 1, pp. 295-307

Liano, K. (1996). Robust error measure for supervised neural network learning with outliers. IEEE Transactions on Neural Networks, vol. 7, pp. 246-250

Pernia-Espinoza, A.V., Ordieres-Mere, J.B., Martinez-de-Pison, F.J., Gonzalez-Marcos, A. (2005). TAO-robust backpropagation learning algorithm. Neural Networks, vol.18, pp. 191-204

Rousseeuw, P.J. (1984). Least median of squares regression. Journal of the American Statistical Association, 79, pp. 871-880

Rousseeuw, P.J. (1985). Multivariate Estimation with High Breakdown Point. Mathematical Statistics and Applications, vol. B, Reidel, the Netherlands

Rousseeuw, P.J., Leroy, A.M. (1987). Robust Regression and Outlier Detection. Wiley, New York

Rusiecki, A.L. (2006). Robust Learning Algorithm with the Variable Learning Rate. ICAISC 2006, Artificial Intelligence and Soft Computing, pp. 83-90

Stromberg, A.J., Ruppert, D. (1992). Breakdown in nonlinear regression. Journal of American Statistical Association, 87, pp. 991–997

Vogl, T.P., et al. (1988). Accelerating the convergence of the backpropagation method. Biological Cybernetics, vol. 59, pp. 256-264

KEY TERMS

Feedforward Neural Networks: Artificial NN consisting of units arranged in layers with only forward connections to units in subsequent layers.

Gross Errors: Large value errors, often caused by human mistakes, measurement errors, etc.

Leverage Points: Grossly aberrant values of measured or assumed system inputs

Outlier: Observation that is significantly different from majority of data

Robust Estimator: Estimator able to classify data into outliers and clean observations, and to find a reasonable fit to the bulk of data.

Robust Learning Algorithm: NN learning algorithm that can act well even if outliers or leverage points are present in training sets

Robust Statistics: Part of statistics developing methods that should give useful results when certain assumptions (for example of iid light tailed errors) are relaxed

R

Rough Set–Based Neuro–Fuzzy System

Kai Keng Ang
Institute for Infocomm Research, Singapore

Chai Quek
Nanyang Technological University, Singapore

INTRODUCTION

Neuro-fuzzy hybridization is the oldest and most popular methodology in soft computing (Mitra & Hayashi, 2000). Neuro-fuzzy hybridization is known as Fuzzy Neural Networks, or Neuro-Fuzzy Systems (NFS) in the literature (Lin & Lee, 1996; Mitra & Hayashi, 2000). NFS is capable of abstracting a fuzzy model from given numerical examples using neural learning techniques to formulate accurate predictions on unseen samples. The fuzzy model incorporates the human-like style of fuzzy reasoning through a linguistic model that comprises of if-then fuzzy rules and linguistic terms described by membership functions. Hence, the main strength of NFS in modeling data is universal approximation (Tikk, Kóczy, & Gedeon, 2003) with the ability to solicit interpretable if-then fuzzy rules (Guillaume, 2001). However, modeling data using NFS involves the contradictory requirements of interpretability versus accuracy. Prevailingly, NFS that focused on accuracy employed optimization which resulted in membership functions that derailed from human-interpretable linguistic terms, or employed large number of if-then fuzzy rules on high-dimensional data that exceeded human level interpretation.

This article presents a novel hybrid intelligent Rough set-based Neuro-Fuzzy System (RNFS). RNFS synergizes the sound concept of knowledge reduction from rough set theory with NFS. RNFS reinforces the strength of NFS by employing rough set-based techniques to perform attribute and rule reductions, thereby improving the interpretability without compromising the accuracy of the abstracted fuzzy model.

BACKGROUND

The core problem in soft computing is about bridging the gap between subjective knowledge and objective data (Dubois & Prade, 1998). There are two approaches of addressing this problem; namely, modeling data in which a function is built to accurately mimic the data, and abstracting data in which a system is built to produce articulated knowledge preferably in natural language form (Dubois & Prade, 1998). The emphasis of the former is on the ability to reproduce what has been observed. Neural networks with their prominent learning capabilities inspired from biological systems are highly suitable in this approach. On the other hand, the emphasis of the latter is on the ability to explain the data in a human interpretable way. Fuzzy systems with the ability of modeling linguistic terms that are expressions of human language are likewise highly effective in this approach. In fuzzy systems, linguistic expressions are formulated from explicit knowledge in the form of if-then fuzzy rules where the linguistic terms of the antecedents and consequents are fuzzy sets. However, the parameters of these linguistic expressions are sometimes difficult to specify and have to be manually tuned. In contrast, although neural networks are capable of learning from data, they are black box models and thus soliciting knowledge from neural networks is not a straightforward task. Hence, a neural network is capable of modeling data, but a user cannot learn from it. On the other hand, a user can learn from a fuzzy system, but it is not capable of learning from data.

Neuro-fuzzy hybridization synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Thus, Neuro-Fuzzy Systems (NFS) are gray-box models that are capable of abstracting a fuzzy model from given numerical examples using neural learning techniques. Hence, a Neuro-Fuzzy System learns and at the same time, a user can learn from it as well. However, the use of NFS in abstracting data involves two contradictory requirements in fuzzy modeling: interpretability versus accu-

racy (Casillas, Cordón, Herrera, & Magdalena, 2003). In practice, only one of the two properties prevails. Hence, they can be classified as *linguistic NFS* that are focused on interpretability, mainly using the Mamdani model (Mamdani & Assilian, 1975); and *precise NFS* that are focused on accuracy, mainly using the Takagi-Sugeno-Kang model (Takagi & Sugeno, 1985).

Prevailing research on modeling data using linguistic NFS focused on increasing accuracy as much as possible but neglected interpretability (Casillas et al., 2003). Existing linguistic NFS such as FALCON (Lin & Lee, 1996), POPFNN (Quek & Zhou, 2001) and GenSoFNN (Tung & Quek, 2002) employ the hybrid learning approach to abstract model from numerical data. In this approach, clustering is used in the first stage to generate the membership functions and competitive learning is used to identify the if-then fuzzy rules; followed by supervised learning that uses backpropagation in the final stage to optimize the membership functions. The unconstrained optimization in the final stage increases the accuracy of the abstracted model, but it resulted in membership functions that are derailed from human-interpretable linguistic terms (de Oliveira, 1999). Although the definition of interpretability and its criteria is subjected to controversial discussion, interpretable linguistic variables is often associated with the shape and mutual overlapping of the membership functions (Mikut, Jakel, & Groll, 2005). Nevertheless, formal definition on the semantic properties of interpretable linguistic variables were proposed (Mikut et al., 2005; de Oliveira, 1999); namely, coverage, normalized, convex and ordered. Interpretability is vital to NFS in modeling data because if neglected, they degenerate into black-box models in which the advantages over other methods such as neural networks are lost (Casillas et al., 2003; Mikut et al., 2005). Therefore, abstracting a fuzzy model that is not humanly interpretable derails the fundamental purpose of using NFS.

In addition, a large number of if-then fuzzy rules are required to model high dimensional data, which in turn exceeds the human interpretation capacity (Casillas et al., 2003). This interpretability issue on large number of if-then rules motivates the complexity reduction of NFS. This is similar to the problems encountered by numerical data driven techniques in data mining (Han & Kamber, 2001). These techniques rely on heuristics to guide or reduce their search space horizontally or vertically (Lin & Cercone, 1997). Horizontal reduction is realized by the merging of identical data tuples

or the quantization of continuous numerical values while vertical reduction is realized by feature selection methods. In Linguistic NFS, the former corresponds to the conversion of numerical inputs from a continuous range to a finite number of linguistic terms using membership functions while the latter corresponds to fuzzy if-then rule pruning and reduction. In some existing linguistic neuro-fuzzy systems, vertical reduction is employed by identifying fewer if-then fuzzy rules using certain heuristic threshold (Quek & Zhou, 2001), or by applying pruning based on certainty factors (Tung & Quek, 2002). However, if the number of if-then rules is bounded as a practical limitation through the use of heuristic thresholds, then the universal approximation property is lost (Moser, 1999).

Recently, rough set theory (Pawlak, 1991), one of the methodologies in soft computing, has shown to provide efficient techniques of finding hidden patterns in data (Pawlak, 2002). Rough set-based methods have shown the potential for feasible feature selection with the ability to significantly reduce the pattern dimensionality in neural networks. This motivates the synergy of rough set-based methods with NFS to increase the interpretability of the abstracted model without compromising the accuracy.

ROUGH SET-BASED NEURO-FUZZY SYSTEM

This article presents the hybrid intelligent Rough set-based Neuro-Fuzzy System (RNFS) (Ang & Quek, 2006a), which synergizes the sound concept of knowledge reduction in rough set theory with the human-like reasoning style of fuzzy systems and the learning and connectionist structure of neural networks. Details on the architecture and learning process of the RNFS are described in the following sections.

Architecture of RNFS

The architecture of the RNFS is a five-layer neural network shown in Figure 1. Its architecture is developed using the Pseudo Outer-Product based Fuzzy Neural Network using the Compositional Rule of Inference and Singleton fuzzifier (POPFNN-CRI(S)) (Ang, Quek, & Pasquier, 2003) as a foundation. For simplicity, only the interconnections for the output y_m are shown. Each layer in RNFS performs a specific

Figure 1. Architecture of rough set-based neuro-fuzzy system
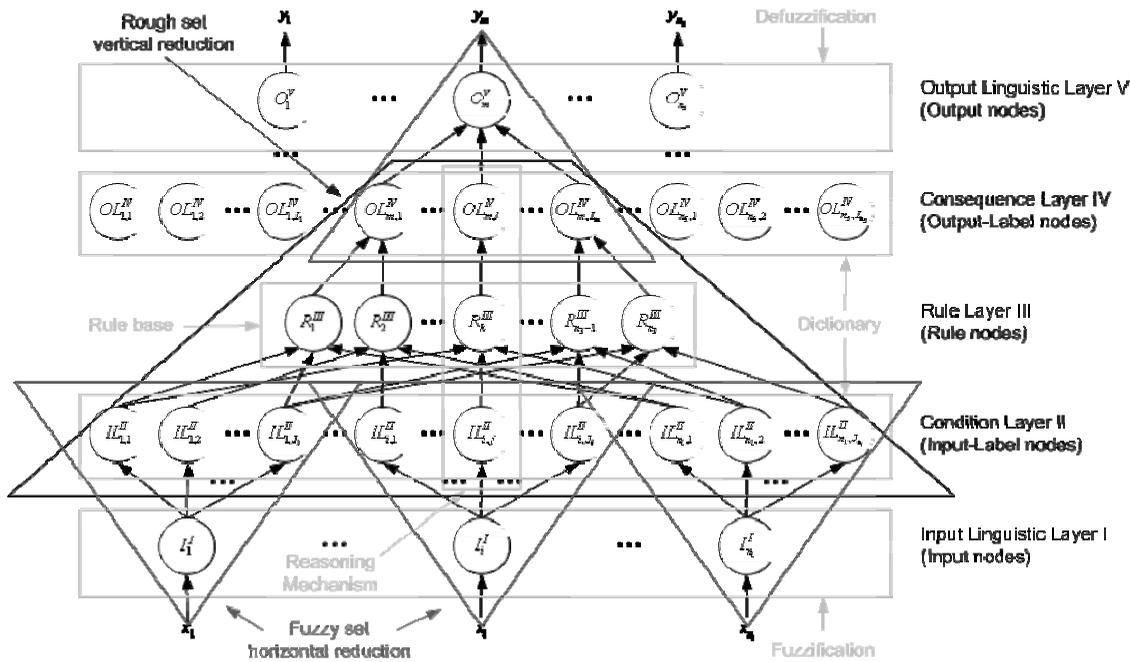
fuzzy operation, so the nodes and operations of each layer are noted with a superscript of *I* to *V* for clarity. The inputs and outputs of the RNFS are represented as non-fuzzy vector $\mathbf{X} = [x_1, x_2,...,x_i,...,x_{n_1}]$ and non-fuzzy vector $\mathbf{Y} = [y_1, y_2,...,y_m,...,y_{n_5}]$ respectively. The fuzzification of the inputs and the defuzzification of the outputs are respectively performed by the input and output linguistic layers respectively while the reasoning mechanism is collectively performed by the condition, rule and consequence layers. Each rule node R_k^{III} in Figure 1 is linked to only one input-label node from each input node and only one output-label node from each output node. The links of rule nodes to the antecedent in the condition layer and the consequent in the consequence layer are mathematically denoted as sets $(\mathbf{C}^I, \mathbf{D}^V)$. The antecedent of rule nodes is represented by a set $\mathbf{C}^I = \{c_1, c_2,...,c_i,...,c_{n_1}\}$ where c_i is a condition variable such that $c_i \in \mathbf{IL}_i^{II}$. The set of condition labels c_i can assume is semantically represented by the set of input-label nodes $\mathbf{IL}_i^{II} = \{IL_{i,1}^{II}, IL_{i,2}^{II},...,IL_{i,j}^{II},...,IL_{i,J_i}^{II}\}$ but a computational notation of $\mathbf{IL}_i^{II} = \{1,2,...,j,...,J_i\}$ is used in this article. Similarly, the consequent of rule nodes is represented by a set $\mathbf{D}^V = \{d_1, d_2,...,d_m,...,d_{n_5}\}$

where d_m is a consequent variable such that $d_m \in \mathbf{OL}_m^{IV}$. The set of consequent labels d_m can assume is semantically represented by the set of output-label nodes $\mathbf{OL}_m^{IV} = \{OL_{m,1}^{IV}, OL_{m,2}^{IV},...,OL_{m,l}^{IV},...,OL_{m,L_m}^{IV}\}$ but a computational notation of $\mathbf{OL}_m^{IV} = \{1,2,..,l,...,L_m\}$ is used in this article. The specific links of a rule node R_k^{III} to the antecedent in the condition layer and the consequent in the consequence layer is denoted as $(\mathbf{C}_k^I, \mathbf{D}_k^V)$.

The novel characteristics of RNFS are:

- *Vertical reduction of if-then fuzzy rules* – is performed in RNFS in which the Rough Set-based Pseudo Outer-Product (RSPOP) algorithm (Ang & Quek, 2005) is used to identify if-then fuzzy rules, perform attribute reduction and rule reduction using rough set-based knowledge reduction. This vertical reduction process is performed autonomously without relying on user-defined heuristic thresholds to identify fewer if-then fuzzy rules.

- *Supervised learning* – is employed in RNFS in which the Supervised Pseudo Self-Evolving Cerebellar (SPSEC) algorithm (Ang & Quek, 2006a) is used to generate membership functions and the RSPOP algorithm is used to identify the if-then

R

fuzzy rules, instead of using backpropagation or hybrid learning approach.

- *Automated model abstraction* – is performed by RNFS without the need of user-defined parameters in the neural learning algorithms that generate the membership functions and identify the if-then fuzzy rules. Thus, it does not require specialized skills and knowledge.

Supervised Learning Process of RNFS

The RNFS employs a novel supervised learning approach that comprises mainly of two algorithms: the SPSEC (Ang & Quek, 2006a) that generates membership functions, and the RSPOP (Ang & Quek, 2005) that identifies the if-then fuzzy rules.

Figure 2 illustrates the neural learning process of the Supervised Pseudo Self-Evolving Cerebellar (SPSEC) membership function generation algorithm (refer to (Ang & Quek, 2006a) for details on the algorithm). Figure 2(a) shows steps 1-2 where SPSEC constructs a cerebellar structure with *m* regularly spaced neurons that spans the input space, in which the cerebellum is the part of our brain that is involved in learning of motor skills and provides precise coordination of motor control for our body parts (Kandel, Schwartz, & Jessell, 1995). These steps model the first-stage development process of our nervous system where the basic architecture and coarse connection patterns are laid out without any activity-dependent processes (Kandel et al., 1995). Figure 2(b) shows steps 3-4, where SPSEC performs structural learning by performing a one-pass weight learning using Gaussian neighborhood learning to determine the distribution of the training data, and pseudo self-evolves this cerebellar structure by identifying surviving neurons with high trophic factor whose weights form a peak while the remaining neurons are removed. These steps model the second-stage development process of our nervous system where initial architecture is refined in activity-dependent ways (Kandel et al., 1995). Figure 2(c) show step 5, where the surviving neurons' weights are the parameters of the resulting Gaussian membership function that reconcile with the semantics of interpretable linguistic variables.

The SPSEC algorithm (Ang & Quek, 2006a) is capable of generating effective membership functions that reconcile with semantics of interpretable linguistic variables; namely, coverage, normalized, convex and

ordered (Mikut et al., 2005). The membership functions generated do not require a further optimization process used in the hybrid learning approach to increase the accuracy of the abstracted model. Eliminating the burden of further optimization process ensures that the membership functions generated do not deviate from human-interpretable terms.

Figure 3 illustrates the rough-set based knowledge reduction process of the Rough set-based Pseudo Outer-Product (RSPOP) if-then fuzzy rule identification algorithm (refer to (Ang & Quek, 2005) for details on the algorithm). Figure 3(a) shows an example of a set of influential if-then fuzzy rules identified by the *RSPOP Rule Identification* steps using Hebbian learning (Hebb, 1949). Figure 3(b) shows the if-then fuzzy rules that are reduced by the *RSPOP Attribute Reduction* steps using rough set-based knowledge reduction (Pawlak, 1991). Figure 3(c) shows the if-then fuzzy rules that are further reduced by the *RSPOP Rule Reduction* steps using rough set-based knowledge reduction (Pawlak, 1991).

The numbers 0-4 in Figure 3 that represent condition and consequent labels in Figure 1 do not show that the if-then fuzzy rules identified are interpretable. To illustrate the intuitiveness of the if-then fuzzy rules identified, an example on the mapping of semantic labels to each condition and consequent labels is illustrated in Table 1.

The example in Figure 3 and Table 1 shows that the RSPOP algorithm is capable of identifying fewer but effective if-then fuzzy rules that facilitates human interpretation. The fewer number of if-then fuzzy rules identified is effective because RSPOP integrates the knowledge reduction technique in rough set theory with the Hebbian learning technique to identify non-redundant if-then fuzzy rules. The accuracy of the abstracted model is not compromised because only reducts that do not deteriorate the accuracy of the abstracted model are reduced by RSPOP.

FUTURE TRENDS

The proposed RNFS architecture is based on the synergy of the sound concept of knowledge reduction in rough set theory with NFS (Ang & Quek, 2005). Existing NFS are not capable of abstracting accurate and interpretable models from high-dimensional data without first performing feature selection to reduce

Figure 2. SPSEC neural learning process (a) Steps 1-2 constructs a basic cerebellar structure consisting of regularly spaced neurons, (b) Steps 3-4 active neurons stablize through the update of trophic factors while competitors die, (c) Steps 5 surviving neurons form the resulting membership functions that reconcile with semantics of linguistic variables

(a)

(b)

(c)
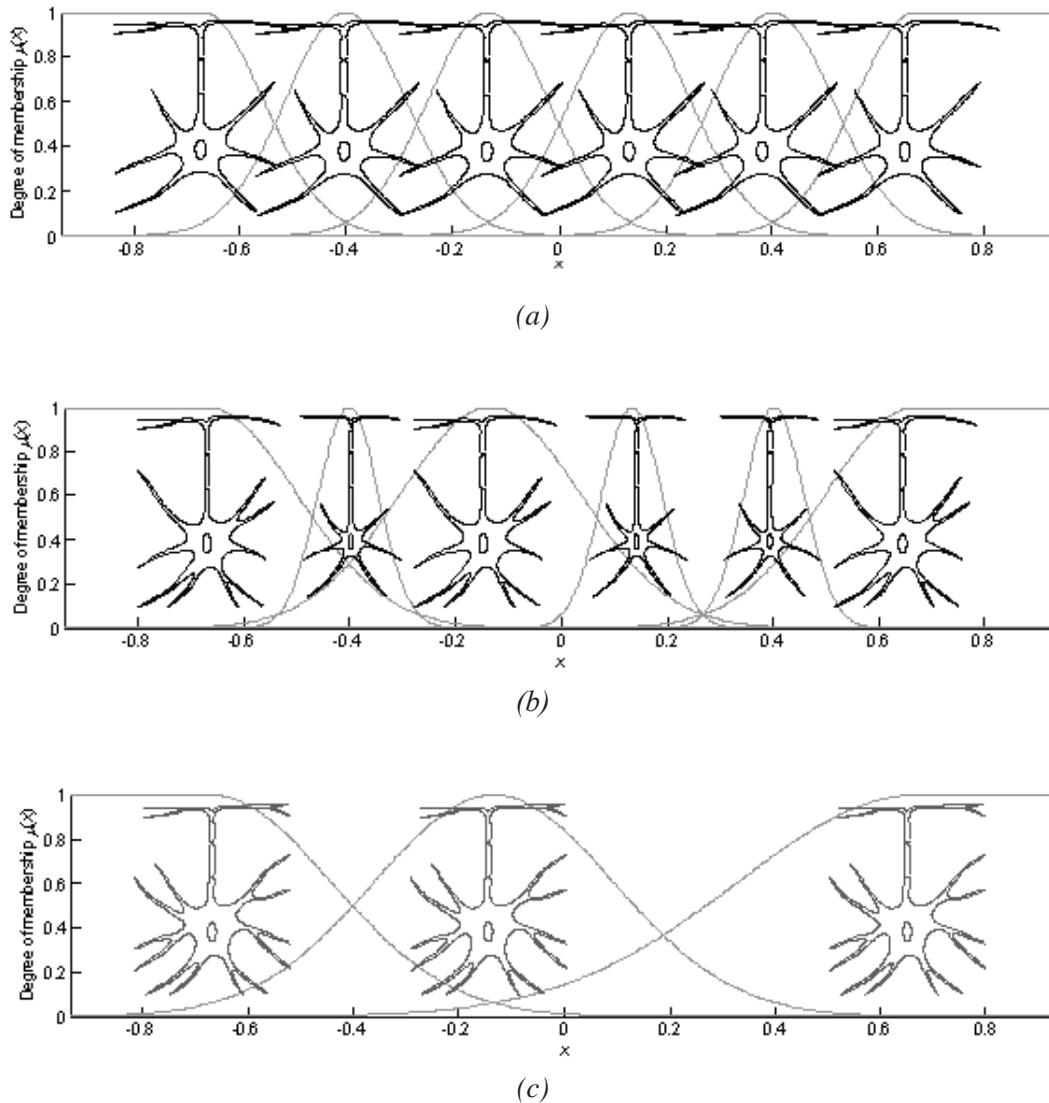
the data dimensionality to a manageable quantity. The synergy of Rough set-based knowledge reduction strengthens and empowers NFS with the potential application on high-dimensional data such as microarray gene expressions and function Magnetic Resonance Imaging (fMRI). Hence, further investigation on the adequacy of abstracting high-dimensional data using RNFS should be carried out and compared against other prevailing approaches.

In addition, existing NFS relied on user-defined parameters to avoid over-fitting the training data in order to generalize well on unseen test data. This issue is known as overfitting avoidance and Occam's razor in pattern recognition literature (Duda, Hart, & Stork, 2001). Although the RNFS is capable of autonomously abstracting a model from numerical data without requiring user-defined parameters, preliminary investigations

Figure 3. RSPOP rule identification process (a) RSPOP identifies an initial set of if-then fuzzy rules identified using Hebbian learning; (b) RSPOP performs attribute reduction on the initial set of if-then fuzzy rules using rough set-based knowledge reduction; and (c) RSPOP performs a further rule reduction on the if-then fuzzy rules

Rules from RSPOP Rule Identification

k	C_a^i				D_a^i
	c_1	c_2	c_3	c_4	d_1
1	2	3	1	1	1
2	2	1	2	1	3
3	1	2	3	1	2
4	4	2	4	1	1
5	3	2	1	2	2
6	3	3	1	2	1
7	1	1	2	2	4
8	3	1	2	2	2
9	2	2	2	2	2
10	2	2	3	2	2
11	4	3	3	2	1
12	3	2	4	2	2
13	1	3	1	3	2
14	4	3	1	3	1
15	1	2	2	3	3
16	3	3	2	3	1
17	3	3	2	3	1
18	1	1	3	3	3
19	2	1	4	3	2
20	2	2	1	4	2
21	1	2	2	4	3
22	4	3	2	4	1

Rules after RSPOP Attribute Reduction

k	C_a^i				D_a^i
	c_1	c_2	c_3	c_4	d_1
1	2	3	0	1	1
2	2	1	0	1	3
3	1	2	0	1	2
4	4	2	0	1	1
5	3	2	0	2	2
6	3	3	0	2	1
7	1	1	0	2	4
8	3	1	0	2	2
9	2	2	0	2	2
10	4	3	0	2	1
11	1	3	0	3	2
12	4	3	0	3	1
13	1	2	0	3	3
14	3	2	0	3	1
15	3	3	0	3	1
16	1	1	0	3	3
17	2	1	0	3	2
18	2	2	0	4	2
19	1	2	0	4	3
20	4	3	0	4	1

Rules after RSPOP Rule Reduction

k	C_a^i				D_a^i
	c_1	c_2	c_3	c_4	d_1
1	2	3	0	1	1
2	2	1	0	1	3
3	1	2	0	1	2
4	4	2	0	1	1
5	3	2	0	2	2
6	3	3	0	0	1
7	1	1	0	2	4
8	3	1	0	2	2
9	2	2	0	2	2
10	4	3	0	0	1
11	1	3	0	3	2
12	1	2	0	3	3
13	3	2	0	3	1
14	1	1	0	3	3
15	2	1	0	3	2
16	2	2	0	4	2
17	1	2	0	4	3

Reduct

Dispensable

(a)   *(b)*   *(c)*

revealed that RNFS may possibly abstract a model that overfits the training data. Therefore, further studies on how to avoid overfitting the training data by RNFS without relying on user-defined parameters should be carried out.

CONCLUSION

This article presents a Rough set-based Neuro-Fuzzy System (RNFS), which is a hybrid intelligent system that synergizes the sound concept of knowledge reduction in rough set theory with the human-like reasoning style of fuzzy systems and the learning and connectionist structure of neural networks. The main strength of Neuro-Fuzzy hybridization is universal approximation

(Tikk et al., 2003) with the ability to solicit interpretable if-then fuzzy rules (Guillaume, 2001). Rough set-based Neuro-Fuzzy hybridization strengthens it further by improving the interpretability as well as the accuracy of existing Neuro-Fuzzy hybridization. Recently, the Rough set-based Neuro-Fuzzy approach of abstracting models from data has been successfully applied to various applications such as traffic flow prediction (Ang & Quek, 2005), financial stock trading (Ang & Quek, 2006b) and the classification of biomedical data (Ang & Quek, 2006a). Hence, the potential of RNFS is exciting as it improves the interpretability of NFS without compromising the accuracy of the abstracted model.

Table 1. Example on the semantic interpretation of the if-then fuzzy rules identified in Figure 3

R_1 :	IF	X_1 is	Medium	AND	X_2 is	High	AND	X_4 is	Low	
R_4 :	IF	X_1 is	Very High	AND	X_2 is	Medium	AND	X_4 is	Low	
R_6 :	IF	X_1 is	High	AND	X_2 is	High				
R_{10} :	IF	X_1 is	Very High	AND	X_2 is	High				
R_{13} :	IF	X_1 is	High	AND	X_2 is	Medium	AND	X_4 is	High	
	THEN Y is Low									
R_3 :	IF	X_1 is	Low	AND	X_2 is	Medium	AND	X_4 is	Low	
R_5 :	IF	X_1 is	High	AND	X_2 is	Medium	AND	X_4 is	Medium	
R_8 :	IF	X_1 is	High	AND	X_2 is	Low	AND	X_4 is	Medium	
R_9 :	IF	X_1 is	Medium	AND	X_2 is	Medium	AND	X_4 is	Medium	
R_{11} :	IF	X_1 is	Low	AND	X_2 is	High	AND	X_4 is	High	
R_{15} :	IF	X_1 is	Medium	AND	X_2 is	Low	AND	X_4 is	High	
R_{16} :	IF	X_1 is	Medium	AND	X_2 is	Medium	AND	X_4 is	Very High	
	THEN Y is Medium									
R_2 :	IF	X_1 is	Medium	AND	X_2 is	Low	AND	X_4 is	Low	
R_{12} :	IF	X_1 is	Low	AND	X_2 is	Medium	AND	X_4 is	High	
R_{14} :	IF	X_1 is	Low	AND	X_2 is	Low	AND	X_4 is	High	
R_{17} :	IF	X_1 is	Low	AND	X_2 is	Medium	AND	X_4 is	Very High	
	THEN Y is High									
R_7 :	IF	X_1 is	Low	AND	X_2 is	Low	AND	X_4 is	Medium	
	THEN Y is Very High									

REFERENCES

Ang, K. K., & Quek, C. (2005). RSPOP: Rough Set-Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. *Neural Computation, 17*(1), 205-243 .

Ang, K. K., & Quek, C. (2006a). Rough Set-based Neuro-Fuzzy System. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '06)* (pp. 742-749).

Ang, K. K., & Quek, C. (2006b). Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach. *IEEE Transactions on Neural Networks, 17*(5), 1301-1315.

Ang, K. K., Quek, C., & Pasquier, M. (2003). POP-FNN-CRI(S): pseudo outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier. *IEEE Transactions on Systems, Man and Cybernetics, Part B, 33*(6), 838-849.

Casillas, J., Cordón, O., Herrera, F., & Magdalena, L. (2003). *Interpretability Issues in Fuzzy Modeling* (Studies in fuzziness and soft computing, No. 128). Berlin: Springer-Verlag.

de Oliveira, J. V. (1999). Towards neuro-linguistic modeling: Constraints for optimization of membership functions. *Fuzzy Sets and Systems, 106*(3), 357-380.

Dubois, D., & Prade, H. (1998). Soft computing, fuzzy logic, and artificial intelligence. *Soft Computing - A Fusion of Foundations, Methodologies and Applications, 2*(1), 7-11.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification* (2nd ed.). New York: John Wiley.

Guillaume, S. (2001). Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems, 9*(3), 426-443.

Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann.

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.

Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (1995). *Essentials of neural science and behavior*. Norwalk, CT: Appleton & Lange.

Lin, C.-T., & Lee, C. S. G. (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems* . Upper Saddle River, NJ: Prentice Hall.

Lin, T. Y., & Cercone, N. (1997). *Rough Sets and Data Mining: Analysis of Imprecise Data*. Boston, London, Dordrecht: Kluwer Academic Publishers.

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies, 7*(1), 1-13.

Mikut, R., Jakel, J., & Groll, L. (2005). Interpretability issues in data-based learning of fuzzy systems. *Fuzzy Sets and Systems, 150*(2), 179-197.

Mitra, S., & Hayashi, Y. (2000). Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks, 11*(3), 748-768.

Moser, B. (1999). Sugeno controllers with a bounded number of rules are nowhere dense. *Fuzzy Sets and Systems, 104*(2), 269-277.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Dordrecht, Boston: Kluwer Academic Publishers.

Pawlak, Z. (2002). Rough sets and intelligent data analysis. *Information Sciences, 147*(1-4), 1-12.

Quek, C., & Zhou, R. W. (2001). The POP learning algorithms: reducing work in identifying fuzzy rules. *Neural Networks, 14*(10), 1431-1445.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics, 15*(1), 116-132.

Tikk, D., Kóczy, L. T., & Gedeon, T. D. (2003). A survey on universal approximation and its limits in soft computing techniques. *International Journal of Approximate Reasoning, 33*(2), 185-202.

Tung, W. L., & Quek, C. (2002). GenSoFNN: a generic self-organizing fuzzy neural network. *IEEE Transactions on Neural Networks, 13*(5), 1075-1086.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-I,II,III. *Information Sciences, 8(3), 199-249; 8(4), 301-357; 9(1)*, 43-80.

R

KEY TERMS

Attribute Reduction: The process whereby dispensable attributes are removed from the knowledge while maintaining knowledge consistency.

Fuzzy System: A system whose variables range over states that are fuzzy sets. A fuzzy system is capable of modelling the linguistic terms that are expressions of human language.

Knowledge Reduction: Knowledge reduction in rough set theory comprises of attribute reduction and partial attribute reduction.

Neural Network: A network of many simple processors called units or neurons. A neural network is capable of learning the nonlinear relationships in data.

Neuro-Fuzzy System: A hybrid intelligent system that synergizes the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks.

Rough Set: A rough set is a formal approximation of a crisp set in terms of a pair of sets that give the lower and upper approximation of the original set

Rough Set-Based Neuro Fuzzy System: A hybrid intelligent system that synergizes the sound concept of knowledge reduction in rough set theory with neuro-fuzzy systems.

Rule Reduction: The process of partial attribute reduction whereby dispensable attributes from certain rules of the knowledge is removed while maintaining knowledge consistency.

Rule Engines and Agent–Based Systems

Agostino Poggi
Università di Parma, Italy

Michele Tomaiuolo
Università di Parma, Italy

INTRODUCTION

Expert systems are successfully applied to a number of domains. Often built on generic *rule-based systems*, they can also exploit optimized algorithms.

On the other side, being based on loosely coupled components and peer to peer infrastructures for asynchronous messaging, *multi-agent systems* allow *code mobility*, adaptability, easy of deployment and reconfiguration, thus fitting distributed and dynamic environments. Also, they have good support for domain specific *ontologies*, an important feature when modelling human experts' knowledge.

The possibility of obtaining the best features of both technologies is concretely demonstrated by the integration of *JBoss Rules*, a rule engine efficiently implementing the *Rete-OO algorithm*, into *JADE*, a *FIPA*-compliant *multi-agent system*.

BACKGROUND

Rule Engines

The advantages of *rule-based systems* over procedural programming environments are well recognized and widely exploited, above all in the context of business applications. Working with rules helps keeping the logic separated from the application code: it can be modified by non-developers and, being centralized in one point, it can be analyzed and validated. *Rule engines* are often well optimized, being able to efficiently reduce the number of rules to match against the updated knowledge base.

Rule-based systems can also be augmented with ideas and techniques developed in other research fields, leading for example to fuzzy rule-based systems, which exploit fuzzy logic to deal with imprecision and uncertainty about the knowledge base. Moreover, sometimes these systems are coupled with genetic algorithms and evolutionary programming to generate complex classifiers.

One of the most notable application of rule-based systems are *expert systems*, where the rule set is a representation of an expert's knowledge. In such systems, the AI (Artificial Intelligence) is supposed to perform in a similar manner to the expert, when exposed to the same data.

Among the different mechanisms to implement a rule-engine, *Rete algorithm* (Forgy, 1982) has gained more and more popularity, mainly thanks to the high degree of optimization that can be obtained. At NASA Johnson Space Center, *Rete algorithm* was implemented in a whole generation of rule engines. OPS5 was soon replaced by its descendant, ART, and in 1984 by the more famous CLIPS.

Nowadays, one of the most widespread engines implementing *Rete* is *Jess* (Friedman-Hill, 2000), at first developed as a Java port of CLIPS at Sandia National Laboratories in late 1990s. *Jess* has also been widely adopted by the agent community to realize rule-based agent systems (Cardoso, 2007).

A different yet promising rule-engine is *JBoss Rules* (Proctor, Neale, Frandsen, Griffith & Tirelli, 2007), formerly *Drools*. It is a quite new, but already well known, freeware tool implementing so-called *Rete-OO algorithm*.

Its open-source availability is a clear advantage over *Jess*, but an even greater advantage is due to the implementation of a particular adaptation of the *Rete algorithm* for the object-oriented world, rather than a literal one. This way, the burden of integrating the rule-engine and application rules with existing external objects is greatly reduced. In fact, *JBoss Rules* uses plain Java objects to represent rules and facts, which can be modified through their public methods and properties. Rules can be specified through an appropriate syntax, or through xml structures, and their conditions and consequences can be expressed using different scripting

R

Figure 1. Basic example from the JBoss Rules handbook

```
package org.drools.examples

import org.drools.examples.HelloWorldExample.Message;

rule "Hello World"
  when
      m : Message( status == Message.HELLO, message : message )
  then
      System.out.println( message );
      m.setMessage( "Goodbye cruel world" );
      m.setStatus( Message.GOODBYE );
      update( m );
end

rule "GoodBye"
  no-loop true
  when
      m : Message( status == Message.GOODBYE, message : message )
  then
      System.out.println( message );
end
```

languages, as Python, Groovy and Java. Instead *Jess* only accepts rules written in the CLIPS language, thus requiring developers to learn a new Lisp-like language and deploy additional efforts to adapt it to their object-oriented development environment.

Agent-Based Systems

Multi-agent systems (MAS) show some complementary features which can be useful in many rule-based application, above all asynchronous interaction protocols and semantic languages. In multi-agent systems, in fact, many intelligent agents interact with each other. The agents are considered to be autonomous entities, and their interactions can be either cooperative or selfish (i.e. they can share a common goal, as in a production line, or they can pursue their own interests, as in an open marketplace).

The *Foundation for Intelligent Physical Agents* (FIPA, 2002) develops open specifications, to support interoperability among agents and agent-based applications. Specifications for infrastructures include a communication language for agents, services for agents, and they anticipate the management of domain-specific

ontologies. A set of application domains is also specified, including personal assistance for travels, network management, electronic commerce, distribution of audio-visual media. At the core of FIPA model there's the communication among agents; in particular it describes how the agents can exchange semantically-meaningful messages with the aim of completing activities required by the overall application.

Various implementations of FIPA-compliant platforms exist (FIPA implementations, 2003). Among them, *JADE* (Bellifemine, Caire, Poggi & Rimassa, 2003) has gained popularity during the years, while more and more core functionalities and third-party plug-ins were being developed. Currently it supports most of the infrastructure related *FIPA* specifications, like transport protocols, message encoding, and white and yellow pages agents. Moreover, it has various tools that ease agent debugging and management.

The possibility of using rules to realize agent systems seems to be promising. On the one hand, rules have been shown suitable to define abstract and real agent architectures and have been used for realizing so-called "*rule-based agents*", that is, agents whose behaviour and/or knowledge is expressed by means of

rules (Shoham, 1993) (Rao, 1996) (Hindriks, de Boer, van der Hoek & Meyer, 1998) (Schroeder & Wagner, 2000). On the other hand, given that rules are easy and suitable means to realize reasoning, learning and knowledge acquisition tasks, rules have been used into so-called "*rule-enhanced agents*", that is, agents whose behaviour is not normally expressed by means of rules, but that use a rule engine as an additional component to perform specific reasoning, learning or knowledge acquisition tasks (Gutknecht, Ferber & Michel, 2000) (Katz, 2002). Both the approaches have some advantages and disadvantages. *Rule-based agents* provide all the advantages of *rule-based systems* and a uniform way to program them, but their performance is inadequate for some kinds of applications. *Rule-enhanced agents* allow the use of different programming paradigms; therefore, it is possible to use the most appropriate paradigm for the realization of the different tasks both to simplify the development and to satisfy the performance requirements, but there is an additional cost for the management of the integration/synchronization of such heterogeneous tasks. With behaviour-based agents, as in *JADE*, the *rule engine* can be integrated into an agent as a behaviour. This approach can alternatively guarantee the advantages of full *rule-based agents* or the ones of *rule-enhanced agents*. In facts, both procedural and rule-based behaviours can be seamlessly added to each deployed agent, according to the application features and requirements.

Code Mobility and Security

Mobile code proves useful in many contexts (Fuggetta, Picco & Vigna, 2000), thanks to its ability to overcome network latency, reduce network load, allow asynchronous execution and autonomy, adapt dynamically, operate in heterogeneous environments, provide robust and fault-tolerant behaviours. *Mobile code* technologies vary from applets and other dynamic code downloading mechanisms, to full mobile agent systems, adhering to models as code on demand, remote evaluation, mobile agents. When a rule engine is integrated into a multi-agent system, two different cases are possible: asking a remote agent to execute a task, or to apply a new rule to its knowledge base. While mobile rules falls into the class of asynchronous requests with deferred execution, instead mobile tasks fall into the synchronous class. In both cases the moved entity is a fragment of code, to

be interpreted by a scripting engine on the target agent, and not a complete thread of execution.

The different *security threats* that a *mobile code* system could face, and the relevant security countermeasures that could be adopted, should also be analyzed. In (Jansen & Karygiannis, 2000) two different classes of attacks are identified, depending on their target: the ones targeting the executing environment of mobile code, and the ones targeting the code itself. While the fact that mobile code could pose threats to its hosting environment is widely accepted, instead often the possibility to face threats against the hosted code is not taken into consideration. This is certainly due to a lack of effective countermeasures to prevent the hosting environment from stealing data and algorithms from the mobile code, from executing it too slowly to be effective, altering its execution flow, or stopping its execution. Experimental algorithms exist to at least detect "a posteriori" this type of threats, including partial result encapsulation, mutual itinerary recording, itinerary recording with replication and voting, execution tracing. Some algorithms even try to prevent some types of attacks to the code hosted in malicious environments, but their real effectiveness has yet to be proved; these include environmental key generation, computing with encrypted functions, and obfuscated code (sometimes called time limited blackbox). On the other hand, potential threats posed by hosted code include masquerading, denial of service, eavesdropping, and alteration. Available security countermeasures to protect the execution environment against potentially malicious mobile code often rely on algorithms to prevent attacks, like software-based fault isolation, safe code interpretation, authorization and attribute certificates, proof carrying code. Other techniques are focused on detecting attacks to the environment and tracing them to their origin; these include state appraisal, signed code, path histories.

Systems based on Java can leverage on the security means provided by the virtual machine, and extend them as needed. In particular, it is possible to define precise protection domains on the basis of *authorization certificates* (Poggi, Tomaiuolo & Vitaglione, 2004). These certificates, attached to mobile code, list a set of granted permissions and are signed by local resource managers. Access rights can also be delegated to other agents, to allow them to complete the requested tasks or to achieve delegated goals (Somacher, Tomaiuolo & Turci, 2002). Finally, masquerading and alteration

threats can be prevented by establishing authenticated, signed and encrypted channels between remote components of the system.

INTEGRATION OF RULES AND AGENTS

Among the different implementations of *rules-enhanced multi-agent systems*, the analysis Drools4JADE (Drools4JADE) can be particularly interesting, as the system resulted from the evaluation of existing technologies in various fields. In fact, the purpose of this project was to not start from scratch, to develop of a totally new agent platform, but instead to build on existing solutions, which already demonstrated to be a sound layer on which more advanced functionalities could be added.

In this case, the chosen agent system is *JADE* (Bellifemine, Caire, Poggi & Rimassa, 2003). Its successful adoption in large international projects, like Agenticities (Poggi, Tomaiuolo & Turci, 2004), openNet and Tech-Net (Willmott, 2004), proved it to be preferable to other solutions, thanks to its simplicity, flexibility, scalability and soundness. As already argued, its integration with an open source object-oriented rule engine, as *JBoss Rules*, in many contexts is to be favoured against the more traditional *JADE-Jess* couple (Cardoso, 2007).

FIPA Interface to the Rule Engine

To the rich features of *JBoss Rules*, an agent environment can add above all the support for communications through ACL (Agent Communication Language) messages, typical of *FIPA* agents. Rules can reference ACL messages in both their precondition and consequence fields. Moreover, a complete support to manipulate facts and rules on rules-enhanced agents through ACL messages can be provided.

Inside the *JBoss Rules* environment a rule is represented by an instance of the Rule class: it specifies all the data of the rule itself, including the pre-conditions making the rule valid and the actions to be performed as consequence of the rule. When a rule is scheduled for execution, i.e. all its preconditions are satisfied by asserted facts, the engine creates a new instance of the embedded scripting environment, set the needed variables inside it and invokes the interpreter to execute the code contained in the consequence section of the rule.

In Drools4JADE, rules-enhanced agents expose a complete API to allow the manipulation of their internal working memory through ACL requests. Their *ontology* defines requests to add rules, assert, modify and retract facts. All these requests must be joined with an *authorization certificate*. Only authorized agents, i.e. the ones that show a certificate listing all needed permissions, can perform requested actions. Moreover, the accepted rules will be confined in a specific protection domain, instantiated according to their own *authorization certificate*.

Security Issues

Mobility of rules and code among agents paves the way for really adaptive applications, but it cannot be fully exploited if security issues aren't properly addressed. The security means implemented in Drools4JADE greatly benefit from the existing infrastructure provided by the underlying Java platform and by *JADE*. The security model of *JADE* deals with traditional user-centric concepts, as principals, resources and permissions. Moreover it provides means to allow delegation of access rights among agents, and the implementation of precise protection domains, by means of authorization certificates.

In the security framework of *JADE*, a principal represents any entity whose identity can be authenticated. Principals are bound to single persons, departments, companies or any other organizational entity. Also single agents are bound to a principal; with respect to his own agents, a user constitutes a parent principal, thus allowing to grant particular permissions to all agents launched by a single user.

Resources that *JADE* security model cares for include those already provided by security Java model (i.e. file system, network connections, environment variables, database connections). Resources typical of multi-agent systems, to be protected against unauthorized accesses, include agents themselves and their executing environment.

A permission represents the capability to perform actions on system resources. To take a decision while trying to access a resource, access control functions compare permissions granted to the principal with permissions required to execute the action; access is allowed if all required permissions are owned.

When an agent is requested to accept a new rule or task, a first access protection involves authenticat-

Figure 2. Actions supported by rules-enhanced agents

code. Agents should be provided means to delegate not only tasks, but even access rights needed to perform those tasks. This is exactly what is made possible through the security package of JADE, where distributed security policies can be checked and enforced on the basis of signed *authorization certificates*.

In this kind of systems, every requested action can be accompanied with a certificate, signed by a local resource manager, listing the permissions granted to the requester. Permissions can be obtained directly from a policy file, or through a delegation process. Through this process, an agent can further delegate a set of permissions to another agent, if it can prove the possession of those permissions.

The final set of permissions received through the request message, can finally be used by the servant agent to create a new protection domain to wrap the mobile code during its execution, protecting the access to system, as well as application, resources.

FUTURE TRENDS

A system architecture founded on *rule engines* and *multi-agent systems* is a good starting point to build a fully distributed environment, where the distributed knowledge can include both data and code. For example it can be used to realize systems for distributed signals and alarms handling, network management etc.

The development of advanced grid features, as transparent and reconfigurable functions for dynamic load balancing, distribution of facts and rules among remote engines, failure detection and recovery, could add even greater value to the system, paving the way for the development of distributed computing environments founded on networks of *FIPA* agents and platforms.

CONCLUSION

ing the requester and checking the authorization to perform the action; i.e.: can the agent really add a new rule, or submit a task, to be performed on its behalf? To perform these tasks, the requester needs particular permissions.

Moreover, to exploit the full power of task delegation and rule mobility, the target agent should be able to restrict the set of resources made accessible to mobile

The integration of an object-oriented *rule engine* and a scripting engine into an agent development framework can provide many advantages. The resulting system joins the soundness of a platform for distributed *multi-agent systems*, with the expressive power of rules and the ability to adapt to changing conditions granted by *mobile code*.

Of course, the development of real world applications poses serious security requirements, which can

be faced by means of detailed security policies and delegation of *authorizations* through signed *certificates*. Application areas include, but certainly are not limited to, e-learning, e-business, service-composition, network management.

REFERENCES

Bellifemine, F., Caire, G., Poggi, A., Rimassa, G. (2003). JADE A White Paper. *Telecom Italia EXP magazine* Vol 3, No 3 September 2003.

Cardoso, H.L. (2007). Integrating JADE and Jess. JADE Web site: http://jade.tilab.com/doc/tutorials/jade-jess/jade_jess.html.

Eberhart. (2002). OntoAgent: A Platform for the Declarative Specification of Agents, In *Proc. of ISWC 2002*, Cagliari, Italy.

FIPA. (2007). FIPA Abstract Architecture Specification. FIPA Web site: http://www.fipa.org/specs/fipa00001/SC00001L.html.

FIPA. (2003). Publicly Available Implementations of FIPA Specifications. FIPA Web site: http://www.fipa.org/resources/livesystems.html.

Forgy, C. L. (1982). Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem, *Artificial Intelligence* 19(1), pp. 17-37.

Friedman-Hill, E.J. (2000). Jess, the Java Expert System Shell. Sandia National Laboratories Web site: http://herzberg.ca.sandia.gov/jess.

Fuggetta, A., Picco, G.P., Vigna, G. (1998). Understanding code mobility, *IEEE Transaction on Software Engineering* 24 (5):342–362.

Gutknecht, O., Ferber, J., Michel, F. (2001). Integrating tools and infrastructures for generic multi-agent systems. In *Proc. of the Fifth Int. Conf. on Autonomous Agents*. Montreal, Canada.

Hindriks, K.V., de Boer, F.S., van der Hoek, W., Meyer, J.C. (1998). Control Structures of Rule-Based Agent Languages. *Proc. ATAL-98*, Paris, France.

Jansen, W., Karygiannis, T. (2000). Mobile agent security. *NIST Special Publication 800-19*.

Katz, E.P. (2002). A Multiple Rule Engine-Based Agent Control Architecture, *Technical Report HPL-2001-283*. HP Laboratories Palo Alto.

Poggi, A., Tomaiuolo, M., Vitaglione, G. 2004. A Security Infrastructure for Trust Management in Multi-agent Systems. *Trusting Agents for Trusting Electronic Societies* 2004: 162-179.

Poggi, A., Tomaiuolo, M., Turci, P. (2004). Using Agent Platforms for Service Composition. ICEIS (4) 2004: 98-105

Proctor, M., Neale, M., Frandsen, M., Griffith, S.Jr., Tirelli, E. (2007). JBoss Rules User Guide. JBoss Web site: http://labs.jboss.com/jbossrules/docs.

Rao, A.S. (1996). AgentSpeak(L): BDI Agent Speak Out in a Logical Computable Language. *Agents Breaking Away*: 42-55.

Schroeder, M., Wagner, G. (2000). Vivid agents: Theory, architecture, and applications. *Int. Journal for Applied Artificial Intelligence*, 14(7): 645-676.

Shoham, Y. (1993) Agent-oriented programming. *Artificial Intelligence*, 60(1): 51-92.

Somacher, M., Tomaiuolo, M., Turci, P. (2002). Goal Delegation in Multiagent System. *AIIA 2002*. Siena, Italy.

Willmott, S. (2004). Deploying Intelligent Systems on a Global Scale. *IEEE Intelligent Systems* Vol. 19(5), September/October 2004: 71-73.

KEY TERMS

Authorization Certificate: A digital document that describes a permission from the issuer to use a service or a resource that the issuer controls or has access to use. Usually it is signed by means of a public key algorithm. The permission in some case can also be delegated.

Expert System: Encodes the knowledge of an expert into the rule set of a rule-based system. When exposed to the same data, the expert system AI will perform in a similar manner to the expert.

Multi-Agent System: A software system based on the interaction of several agents. Such agents could not have all data or all resources needed to achieve an

objective and need to collaborate with other agents. In this case, data is decentralized and execution is asynchronous. Earlier, related fields include Distributed Artificial Intelligence (DAI) and distributed problem solving (DPS).

Ontology: An explicit specification of a conceptualization, formally describing the entities involved in a particular domain and the relationships among them.

Production System (or production rule system): a rule-based system whose rules (termed productions) consist of two parts: a sensory precondition (or "if" statement) and an action (or "then"). If a production's precondition (left-hand side or LHS) matches the current state of the world, then the production is said to be triggered. If a production's action is executed, it is said to have fired. The rule interpreter must provide a mechanism for prioritizing productions when more than

one is triggered. Rule interpreters generally execute a forward chaining algorithm for selecting productions to execute.

Rule-Based System: Created using a set of assertions, which collectively form the "working memory", a database which maintains data about current state or knowledge, a set of rules, specifying how to act on the assertion set, and a rule-engine or interpreter. Basically, rule-based systems can consisting of little more than a set of if-then statements, but provide the basis for so-called "expert systems".

Software Agent: A software entity being able to act with a certain degree of autonomy, in order to accomplish tasks on behalf of its user. While objects are defined in terms of methods and attributes, agents are defined in terms of their behaviours. Usually agents show persistence, autonomy, social ability, reactivity.

Sequence Processing with Recurrent Neural Networks

Chun-Cheng Peng
University of London, UK

George D. Magoulas
University of London, UK

INTRODUCTION

Sequence processing involves several tasks such as clustering, classification, prediction, and transduction of sequential data which can be symbolic, non-symbolic or mixed. Examples of symbolic data patterns occur in modelling natural (human) language, while the prediction of water level of River Thames is an example of processing non-symbolic data. If the content of a sequence will be varying through different time steps, the sequence is called *temporal* or *time-series*. In general, a temporal sequence consists of nominal symbols from a particular alphabet, while a time-series sequence deals with continuous, real-valued elements (Antunes & Oliverira, 2001). Processing both these sequences mainly consists of applying the current known patterns to produce or predict the future ones, while a major difficulty is that the range of data dependencies is usually unknown. Therefore, an intelligent system with memorising capability is crucial for effective sequence processing and modelling.

A recurrent neural network (RNN) is an artificial neural network in which self-loop and backward connections between nodes are allowed (Lin & Lee 1996; Schalkoff, 1997). Comparing to feedforward neural networks, RNNs are well-known for their power to memorise time dependencies and model nonlinear systems. RNNs can be trained from examples to map input sequences to output sequences and in principle they can implement any kind of sequential behaviour. They are biologically more plausible and computationally more powerful than other modelling approaches, such as Hidden Markov Models (HMMs), which have non-continuous internal states, feedforward neural networks and Support Vector Machines (SVMs), which do not have internal states at all.

In this article, we review RNN architectures and we discuss the challenges involved in training RNNs for sequence processing. We provide a review of learning algorithms for RNNs and discuss future trends in this area.

BACKGROUND

One of the first RNNs was the *avalanche network* developed by Grossberg (1969) for learning and processing an arbitrary spatiotemporal pattern. Jordan's *sequential network* (Jordan, 1986) and Elman's *simple recurrent network* (Elman, 1990) were proposed later.

The first RNNs did not work very well in practical applications, and their operation was poorly understood. However, several variants of these models were developed for real-world applications, such as robotics, speech recognition, music composition, vision, and their potential for solving real-world problems has motivated a lot of research in the area of RNNs.

Current research in RNNs has overcome some of the major drawbacks of the first models. This progress has come in the form of new architectures and learning algorithms, and has led in a better understanding of the RNNs' behaviour.

ARCHITECTURES OF RECURRENT NETWORKS

In the literature, several classification schemes have been proposed to organise RNN architectures starting from different principles for the classification, i.e. some consider the loops of nodes in the hidden layers, while others take the types of output into account. For example, they can be organised into *canonical* RNNs and *dynamic MLPs* (Tsoi, 1998a); *autonomous converging* and *non-autonomous non-converging* (Bengio et al., 1993); *locally* (receiving feedback(s) from the

same or directly connected layer), *output feedback*, and *fully connected* (i.e. all nodes are capable to receive and transfer feedback signals to the other nodes, even within different layers) RNNs (dos Santos & Zuben, 2000); *binary* and *analog* RNNs (Orponen, 2000).

From mathematical point of view (Kremer, 2001), assuming that *y* and *z* are respectively the response of the output layer and the output of the hidden layer, a static feedforward neural network can be formulated as follows:

$$y = \phi\left(\mathbf{W}^{\mathrm{II}}z + b^{\mathrm{II}}\right) \quad (1)$$

$$z = \phi\left(\mathbf{W}^{\mathrm{I}}x + b^{\mathrm{I}}\right), \quad (2)$$

where $f(\cdot)$ denotes nonlinear activation function, $\mathbf{W}^{\mathrm{I}}$ and $\mathbf{W}^{\mathrm{II}}$ the weights of the hidden layer and the output layer, *x* the input vector, and *b* the biases. This general form could be easily transformed to describe a Feed-Forward Time-Delayed (FFTD) RNN by substituting the following delayed equations with time index *t*,

$$y(t) = \phi\left(\mathbf{W}^{\mathrm{II}}z(t) + b^{\mathrm{II}}\right) \quad (3)$$

$$z(t) = \phi\left(\mathbf{W}^{\mathrm{I}}s(t) + b^{\mathrm{I}}\right) \quad (4)$$

$$s(t) = \left\{x(t) \oplus x(t-1) \oplus \cdots \oplus x(t-d)\right\}, \quad (5)$$

where *s(t)* denotes the state vector at time *t*, $\oplus$ the Cartesian product, *d* the number of delays. By adding a feedback connection from the hidden layer to the delay unit then Eq. (4) can be stated as

$$z(t) = \phi\left(\Lambda z(t-1) + \mathbf{W}^{\mathrm{I}}x(t) + b^{\mathrm{I}}\right), \quad (6)$$

where Λ is a diagonal matrix, which describes an Elman-type RNN.

For the Nonlinear Autoregressive Network with Exogenous Inputs (NARX) the state is described as

$$s(t) = \left\{x(t) \oplus x(t-1) \oplus \cdots \oplus x(t-d+1)\right\} \oplus$$
$$\left\{y(t-1) \oplus y(t-2) \oplus \cdots \oplus y(t-m)\right\}, \quad (7)$$

where *m* is the number of output feedbacks. The formulations of a fully RNN can also be derived by combining Eqs. (3) and (7) with the following one:

$$z(t) = \phi\left(\Lambda z(t-1) + \mathbf{W}^{\mathrm{I}}s(t) + \mathbf{W}^{\mathrm{I}}x(t) + b^{\mathrm{I}}\right). \quad (8)$$

Table 1 provides an overview of the various architectures and of the relevant literature.

LEARNING ALGORITHMS FOR RECURRENT NETWORKS

With regards to training RNNs and storing information in their internal representations, Gradient Descent-based learning algorithms (GD) are the most commonly applied methods, even though it has been claimed that GD has some drawbacks (Bengio et al., 1994). Firstly, when the delays or recursive connections are very deep, i.e. when long-term memory is required, the backpropagation error may be vanished and the training process could become inefficiently. Secondly, the most common way to apply GD algorithms into RNN is to unfold the recursive layers and train the whole network as a feedforward network. Another drawback is that the generalisation is highly affected by the samples in the training dataset. In temporal processing it is difficult to extract or prepare negative samples from a given

Table 1. Classification summary of RNNs

Recurrence	Globally	Locally	Fully	Partially
Reference	Brouwer (2005) Kremer & Kolen (2000) Puskorius & Feldkamp (1994)	Assaad et al. (2005) Boné & Cardot (2005) Sperduti & Starita (1997) Temurtasm et al. (2004) Tiňo & Mills (2005)	Pedersen (1997)	All, except Pedersen (1997)
Equations	(3),(4),(7)	(3),(6)	(3),(7),(8)	Globally/Locally

training dataset and the specific RNN then predicts or classifies new coming samples according to the learned knowledge only.

In (Bengio et al., 1993), besides Backpropagation Through Time (BPTT) and real-time gradient computation, approaches with space and/or time locality are also reviewed. However, local algorithms can be applied to some specific local feedback RNNs and for short-term memorisation only due to their inherent representation capabilities. The inefficiency of GD in learning long-term dependencies is mainly because previous information is treated initially as noise and gradually is ignored (Bengio et al., 1993; Bengio et al., 1994). Therefore two alternative algorithms are revised and discussed in Bengio's works: the time-weighted pseudo-Newton and the discrete error propagation. The former applies the unfolding method to the pseudo-Newton optimisation and the later considers the limited case of propagation only; it has to be verified whether this would work on other more general situation or not.

Two types of learning algorithms are discussed in (Pearlmutter, 1995): the *fixed point*, and the *nonfixed point*. Well-known algorithms such as BPTT and Real-time Recurrent Learning (RTRL) are included in this classification and a way of introducing time constants and time delays is also suggested. The method of extended RTRL (eRTRL) is also discussed and other relevant approaches, such as Elman nets, Jordan nets, the moving targets method, feedforward nets with state, teach forcing in continuous time and Kalman filter are reviewed. Pearlmutter (1995) also compares the complexity both in time and space, and discusses the learning mode, stability and locality of these algorithms.

For fully connected hidden layer networks and dynamic MLPs, Tsoi (1998b) has investigated two first-order gradient learning algorithms. This work discusses some drawbacks of these methods, such as slow convergence and generalisation, and derives two 2nd-order approaches to speed up the convergence and to tackle the issue of weight pruning; it also provides a discussion on output sensitivity. The lower sensitivity of output to a specific adjustable parameter, the better performance of the network is. Although the related formulas are well defined in this work (Tsoi, 1998b), there is still a crucial constant which is used to set the level of sensitivity that should be defined by the users. Quasi-2nd order methods, such as conjugate gradient, scaled conjugate gradient and Newton approach, have

been also mentioned and pointed out as suitable only for batch training, while Kalman filter and extended Kalman filter are classified as 2nd-order GD based learning algorithms, which can be used under online mode, where extended Kalman filter could be used to prune weights from a RNN.

Kremer (2001) reviews 14 kinds of memories used in spatiotemporal connectionist networks, capable of computing the state vectors, and provides a general formulation for computing output vectors. The author also summaries 10 different kinds of updating rules, such as full GD, truncated GD, autoassociative GD, and stack learning. It examines three open issues: the temporal credit assignment, the representation capabilities and the knowledge encoding.

From the point of view of time-series modelling, Kolehmainen (2003) covers BPTT and RTRL for learning RNNs, while Dietterich (2002) suggests BPTT. In the same vein with Baldi (1993) and Pearlmutter (1995), fixed point networks are also considered and five relative algorithms, such as BPTT and GD learning of time constants, gains and delays are summarised.

Most RNN applications are still using first-order learning algorithms despite the drawbacks of the GD. Some attempts have been made to propose second-order learning algorithms, e.g. dos Santos & von Zuben (2000) proposed a quasi 2nd-order method. Also, simulated annealing has given some promising results but the training time is relatively higher (Bengio et al., 1994).

Table 2 provides an overview of RNNs learning, giving examples of training algorithms for locally and globally RNNs for various applications.

FUTURE TRENDS

Recent directions in RNN research focus on investigating and proposing new ways for better modelling of non-stationarity in sequences, such as sequences produced when modelling speech or handwritten characters, with no temporal independence assumptions.

With regards to architectures, hybrid models based on combinations of Hidden Markov Models and RNNs as well as modular structures are considered promising approaches to solve sequence processing problems that occur in natural language and speech processing. In addition, a number of applications of the so-called Long Short-Term Memory RNN (Hochreiter & Schmidhuber, 1997) have provided some encouraging

Table 2. Recurrent neural networks applications and learning algorithms

Recurrent	Applications	Algorithms*	Typical Problems Encountered	Reference
	Time series prediction: sunspots, Mackey-Glass, laser, reservoir inflows	BPTT, RTRL, DEKF	• Limited choice of cost functions • Occasional instability in the convergence of GD[1] • Prior knowledge required about system tuning • Oversized architecture, poor generalisation	Assaad et al., 2005; Boné & Cardot, 2005; Boné et al., 2002; Pérez-Ortiz et al., 2001
	Symbolic transduction and prediction: grammatical inference	BPTT, RTRL, DEKF	• High computational complexity • Not suitable for complex sequences[2]	Kremer & Kolen, 2000 Pérez-Ortiz et al., 2001
Locally	Classification of structures	BP, eBPTT, eRTRL, rBP	• Prior knowledge required	Brouwer, 2005; Seow & Asari, 2006; Sperduti & Starita, 1997; Xiangrui & Chaudhari, 2004;
	System Identification	Casual rBP	• Slightly higher computational complexity	Campolucci et al., 1999
	Signal processing	Rprop, COM	• Drawbacks of GD methods	Franklin & Locke, 2004; Temurtasm et al., 2004;
	Moore Machine	SpikeProp-TT	• Discontinuous error surfaces • Memory vanishing • Suitable for simpler patterns	Tiño & Mills, 2005
Globally	Time series generation/ prediction	DEKF	• Suitable for short-term and noiseless patterns	Priel & Kanter, 2003; Puskorius & Feldkamp, 1994
	Sequences classification	BP	• Drawbacks of GD methods	Brouwer, 2005

*The notation used here is: BP- backpropagation; BPTT- BP through time; eBPTT- extended BPTT; RTRL- real time recurrent learning; eRTRL- extended RTRL; rBP- recurrent/recursive BP; COM- combination of gradient descent, truncated BPTT and RTRL; DEKF- Decouple Extended Kalman Filter; Rprop- Resilient BP; SpikeProp TT- Spike Propagation Through Time

[1] This can be due to sensitivity to initial conditions and the multitudes of local minima

[2] For example sequences in the area of natural language processing

results, demonstrating that these recurrent architectures can overcome several of the fundamental problems of traditional RNNs, and efficiently learn to solve many previously unlearnable tasks.

As far as RNN training is concerned and despite the popularity of gradient descent approaches, which enforce the monotone decrease of the learning error, there are new learning algorithms that are based on evolutionary algorithms (Schmidhuber et al., 2007) and nonmonotone learning strategies (Peng and Magoulas, 2007) that have shown potential for effective RNN training.

CONCLUSION

Recurrent networks constitute an elegant way of increasing the capacity of feedforward networks to deal with complex data in the form of sequences of patterns. Recurrent neural networks are well known for their power to model temporal dependencies and process sequences for classification, recognition, and transduction. Modern RNNs architectures are capable of learning to solve many previously unlearnable tasks, even in partially observable environments. In this article, we presented several RNN models. We identified the main challenges involved in training RNNs and discussed several algorithmic approaches for training RNN for sequence processing. Lastly, we presented some future directions for work in this area.

REFERENCES

Antunes, C.M. & Oliveira, A.L. (2001). Temporal data mining: an overview. *Proc. KDD Workshop on Temporal Data Mining,* 1-13, San Francisco, CA. 26 August 2001.

Assaad, M., Boné, R. & Cardot, H. (2005). Study of the behavior of a new boosting algorithm for recurrent neural networks. In: Duch W. et al. (eds.) *Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN)*, 169-174.

Baldi, P. (1993). Gradient descent learning algorithm overview: a general dynamical systems perspective. *IEEE Transactions on Neural Networks*, 6(1), 182-195.

Bengio, Y., Frasconia, P. & Gori, M. (1993). Recurrent neural networks for adaptive temporal processing. *Proceedings of the 6th Italian Workshop on Parallel Architectures and Neural Networks*, 85-117.

Bengio, Y., Simard, P. & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157-166.

Boné, R. & Cardot, H. (2005). Time delay learning by gradient descent in recurrent neural networks. In: Duch W. et al. (eds.) *Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN)*, 175-180.

Boné, R., Crucianu, M. & de Beauville, J.-P.A. (2002). Learning long-term dependencies by the selective addition of time-delayed connections to recurrent neural networks. *Neurocomputing*, 48, 251-266.

Brouwer, R.K. (2005). Training of a discrete recurrent neural network for sequence classification by using a helper FNN. *Soft Computing*, 9, 749-756.

Campolucci, P., Uncini, A., Piazza, F. & Rao, B.D. (1999). On-line learning algorithms for locally recurrent neural networks. *IEEE Transactions on Neural Networks*, 10(2), 253-271.

Dietterich, T.G. (2002). Machine learning for sequential data: a review. *Lecture Notes in Computer Science*, 2396.

dos Santos, E.P. & von Zuben, F.J. (2000). Efficient second-order learning algorithms for discrete-time recurrent neural networks. In: L.R. Medsker & L.C. Jain (eds.): *Recurrent Neural Networks: Design and Applications*. CRC Press. New York, 47-75.

Elman, J.L. (1990). Finding structure in time. *Cognitive Science*, 14, 179-211.

Franklin, J.A. & Locke, K.K. (2004). Recurrent neural networks for musical pitch memory and classification. *International Journal on Artificial Intelligence Tools*, 14(9), 329-342.

Grossberg, S. (1969). Some networks that can learn, remember, and reproduce any number of complicated space-time patterns. *International Journal of Mathematics and Mechanics*, 19, 53-91.

S

Hochreiter S. & Schmidhuber J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780.

Jordan, M.I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eight Annual Conference of the Cognitive Science Society*, 531-546.

Kolehmainen M. (2003). *Lecture notes on time-series modeling using neural networks*. University of Kuopio.

Kremer, S.C. & Kolen, J.F. (2000). Dynamical Recurrent Networks for Sequential Data Processing. In: S. Wermter & R. Sun (eds.) Hybrid Neural Systems, Revised Papers From A Workshop (December 04-05, 1998), *Lecture Notes In Computer Science 1778*, London: Springer-Verlag, 107-122.

Kremer S.C. (2001). Spatiotemporal connectionist networks: a taxonomy and review. *Neural Computation*, 13, 249-306.

Lin, C.-T. & Lee, C.S.G. (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. New Jersey: Prentice Hall.

Orponen, P. (2000). An overview of the computational power of recurrent neural networks. *Proceedings of the 9th Finnish AI Conference, Vol. 3: "AI of Tomorrow"*, 89-96.

Pearlmutter, B.A. (1995). Gradient calculations for dynamic recurrent neural networks: a survey. *IEEE Transactions on Neural Networks*, 6(5), 1212-1228.

Pedersen, M.W. (1997). Optimization of recurrent neural networks for time series modeling. *PhD thesis*. Technical University of Denmark.

Peng C.-C. & Magoulas G.D. (2007). Adaptive self-scaling nonmonotone BFGS training algorithm for recurrent neural networks. In J. Marques de Sá, et al., (Eds.), *Lecture Notes in Computer Science, Vol. 4668,* (pp. 259-268). The Proceedings of the 17th International Conference on Artificial Neural Networks (ICANN07) Part I, 9-13 September, Porto, Portugal, Lecture Notes in Computer Science

Pérez-Ortiz, J.A., Calera-Rubio, J. & Forcada, M.L. (2001). Online symbolic-sequence prediction with discrete-time recurrent neural networks. In: Dorffner G. Bischof H. & Hornik K. (eds.) International Conference on Artificial Neural Networks 2001, *Lecture Notes in Computer Science*, 2130, 719-724.

Priel, A. & Kanter, I. (2003). Time series generation by recurrent neural networks. *Annals of Mathematics and Artificial Intelligence*, 39, 315-332.

Puskorius, G.V. & Feldkamp, L.A. (1994). Neuro-control of nonlinear dynamical systems with Kalman filter trained recurrent networks. *IEEE Transactions on Neural Networks*, 5(2), 279-297.

Schalkoff, R.J. (1997). *Artificial Neural Networks*, New York: McGraw-Hill.

Schmidhuber J., Wierstra D., Gagliolo M., & Gomez F. (2007). Training Recurrent Networks by Evolino. *Neural Computation,* 19(3), 757-779.

Seow, M.J. & Asari, V.K. (2006). Recurrent neural network as a linear attractor for pattern association, *IEEE Transactions on Neural Networks*, 17(1), 246-250.

Sperduti, A. & Starita, A. (1997). Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3), 714-735.

Temurtasm, F., Yumusak, N., Gunturkun, R. Temurtas, H. & Cerezci, O. (2004). Elman's recurrent neural networks using resilient back propagation for harmonic detection. In: Zhang C. et al. (eds.): Proceedings of *Pacific Rim International Conference on Artificial Intelligence*, 422-428.

Tiňo, P. & Mills, A. (2005). Learning beyond finite memory in recurrent networks for spiking neurons. In: Wang L. et al. (eds.): Proceedings of International Conference on Natural Computation 2005, *Lecture Notes in Computer Science*, 3611, 666-675.

Tsoi, A.C. (1998a). Recurrent neural network architectures: an overview. In: Giles C.L. & Gori M. (eds.): *Adaptive processing of sequences and data structures*. Berlin: Springer-Verlag, 1-26.

Tsoi A.C. (1998b). Gradient based learning algorithms. In: Giles C.L. & Gori M. (eds.): *Adaptive processing of sequences and data structures*. Berlin: Springer-Verlag, 27-62.

Xiangrui, W. & Chaudhari, N.S. (2004). Recurrent neural networks for learning mixed k^{th}-order Markov chains. In Pal N.R et al. (eds.): *Proceedings of International Conference on Neural Information Processing*, 477-482.

KEY TERMS

Artificial Neural Network: A network of many simple processors, called "units" or "neurons", which provides a simplified model of a biological neural network. The neurons are connected by links that carry numeric values corresponding to weightings and are usually organised in layers. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Backpropagation through Time: An algorithm for recurrent neural networks that uses the gradient descent method. It attempts to train a recurrent neural network by unfolding it into a multilayer feedforward network that grows by one layer for each time step, also called unfolding of time.

Extended Kalman Filter: An online learning algorithm for determining the weights in a recurrent network given target outputs as it runs. It is based on the idea of Kalman filtering, which is a well-known linear recursive technique for estimating the state vector of a linear system from a set of noisy measurements.

Gradient Descent: A popular training algorithm that minimises the total squared error of the output computer by a neural network. To find a local minimum of the error function using gradient descent, one takes steps proportional to the negative of the gradient (or the approximate gradient) of the function at the current point.

Neural Architecture: Particular organisation of artificial neurons and connections between them in an artificial neural network.

Real-Time Recurrent Learning: A general approach to training an arbitrary recurrent network by adjusting weights along the error gradient. This algorithm usually requires very low learning rates because of the inherent correlations between successive node outputs.

Recurrent Neural Network: An artificial neural network with feedback connections. This is in contrast to what happens in a feedforward neural network, where the signal simply passes from the input neurons, through the hidden neurons, to the outputs nodes

Sequence Processing: A sequence is an ordered list of objects, events or data items. Processing of a sequence may involve one or a number of operations, such as classification of the whole sequence into a category; transformation of a sequence into another one; prediction or continuation of a sequence; generation of an output sequence from a single input.

Training Algorithm: A step-by-step procedure for adjusting the connection weights of an artificial neural network. In supervised training, the desired (correct) output for each input vector of a training set is presented to the network, and many iterations through the training data may be required to adjust the weights. In unsupervised training, the weights are adjusted without specifying the correct output for any of the input vectors.

Shortening Automated Negotiation Threads via Neural Nets

Ioanna Roussaki
National Technical University of Athens, Greece

Ioannis Papaioannou
National Technical University of Athens, Greece

Miltiades Anagnostou
National Technical University of Athens, Greece

INTRODUCTION

In the artificial intelligence domain, an emerging research field that rapidly gains momentum is *Automated Negotiations* (Fatima, Wooldridge, & Jennings, 2007) (Buttner, 2006). In this framework, building intelligent agents (Silva, Romão, Deugo, & da Silva, 2001) adequate for participating in negotiations and acting autonomously on behalf of their owners is a very challenging research topic (Saha, 2006) (Jennings, Faratin, Lomuscio, Parsons, Sierra, & Wooldridge, 2001). In automated negotiations, three main items need to be specified (Faratin, Sierra, & Jennings, 1998) (Rosenschein, & Zlotkin, 1994): (i) the negotiation protocol & model, (ii) the negotiation issues, and (iii) the negotiation strategies that the agents will employ.

According to (Walton, & Krabbe, 1995), "*Negotiation is a form of interaction in which a group of agents, with conflicting interests and a desire to cooperate try to come to a mutually acceptable agreement on the division of scarce resources*". These resources do not only refer to money, but also include other parameters, over which the agents' owners are willing to negotiate, such as product quality features, delivery conditions, guarantee, etc. (Maes, Guttman, & Moukas, 1999) (Sierra, 2004). In this framework, agents operate following predefined rules and procedures specified by the employed negotiation protocol (Rosenschein, & Zlotkin, 1994), aiming to address the requirements of their human or corporate owners as much as possible. Furthermore, the negotiating agents use a reasoning model based on which their responses to their opponent's offers are formulated (Muller, 1996). This policy is widely known as the negotiation strategy of the agent (Li, Su, & Lam, 2006).

This paper elaborates on the design of negotiation strategies for autonomous agents. The proposed strategies are applicable in cases where the agents have strict deadlines and they negotiate with a single party over the value of a single parameter (*single-issue bilateral negotiations*). Learning techniques based on MLP and GR Neural Networks (NNs) are employed by the client agents, in order to predict their opponents' behaviour and achieve a timely detection of unsuccessful negotiations. The proposed NN-assisted strategies have been evaluated and turn out to be highly effective with regards to the duration reduction of the negotiation threads that cannot lead to agreements.

The rest of the paper is structured as follows. In the second section, the basic principles of the designed negotiation framework are presented, while the formal problem statement is provided. The third section elaborates on the NN-assisted strategies designed and provides the configuration details of the NNs employed. The fourth section presents the experiments conducted, while the fifth section summarizes and evaluates the results of these experiments. Finally, in the last section, conclusions are drawn and future research plans are exposed.

THE AUTOMATED NEGOTIATION FRAMEWORK BASICS

This paper studies a single issue, bilateral automated negotiation framework. Thus, there are two negotiating parties (Client and Provider) that are represented by mobile intelligent agents. The agents negotiate over a single issue based on an alternating offers protocol

(Kraus, 2001) aiming to maximize the utilities of the parties they represent.

We hereafter consider the case where the negotiation process is initiated by the Client Agent (CA) that sends to the Provider Agent (PA) an initial Request for Proposal (RFP) specifying the features of the service/product its owner is interested to obtain. Without loss of generality, it is assumed that the issue under negotiation is the price of the product or service. Thus, the PA negotiates aiming to agree on the maximum possible price, while the CA aims to reduce the agreement price as much as possible. Once the PA receives the RFP of the CA, it either accepts to be engaged in the specific negotiation thread and formulates an initial price offer, or rejects the RFP and terminates the negotiation without a proposal. At each round, the PA sends to the CA a price offer, which is subsequently evaluated by the CA against its constraints and reservation values. Then, the CA generates a counter-offer and sends it to the PA that evaluates it and sends another counter-offer to the CA. This process continues until a mutually acceptable offer is proposed by one of the negotiators, or one of the agents withdraws from the negotiation (e.g. in case its time deadline is reached without an agreement being in place). Thus, at each negotiation round, the agents may: (i) accept the previous offer, if their constraints are addressed, (ii) generate a counter-offer, or (iii) withdraw from the negotiation.

Quantity p_l^a denotes the price offer proposed by negotiating agent a during negotiation round l. A price proposal p_l^b is always rejected by agent a if $p_l^b \notin \left[p_m^a, p_M^a \right]$, where $\left[p_m^a, p_M^a \right]$ denotes agent-a's acceptable price interval. In case an agreement is reached, we call the negotiation successful, while in case one of the negotiating parties quits, it is called unsuccessful. In any other case, we say that the negotiation thread is active. The objective of our problem is to predict the PA's behaviour in the future negotiation rounds until the CA's deadline expires. More specifically, the negotiation problem studied can formally be stated as follows:

Given: (i) two negotiating parties: a Provider that offers a specific good and a Client that is interested in this good's acquisition, (ii) the acceptable price interval $\left[p_m^C, p_M^C \right]$ for the Client, (iii) a deadline T_C up to which the Client must have completed the negotiation with the Provider, (iv) the final negotiation round index L_C for the Client, (v) a round threshold L_C^d until which the Client must decide whether to continue be-

ing engaged in the negotiation thread or not, and (vi) the vector $P_l^P = \left\{ p_l^P \right\}$, where $l = 2k - 1$ and $k = 1, \ldots, \left\lfloor \frac{L_C^d}{2} \right\rfloor$, of the prices that were proposed by the Provider during the initial $L_C^d - 1$ negotiation rounds, *find* (i) the vector $P_{l'}^P = \left\{ p_{l'}^P \right\}$, where $l' = 2k' - 1$ and $k' = \left\lfloor \frac{L_C^d}{2} \right\rfloor + 1, \ldots, L_C$, of the prices that will be proposed by the Provider during the last $L_C - L_C^d$ rounds, and (ii) decide on whether the Client should continue being engaged in the specific negotiation thread or not.

A NEGOTIATION STRATEGY BASED ON NEURAL NETWORKS

The policy employed by negotiating agents in order to generate a new offer is called *negotiation strategy*. In principle, three main families of automated negotiation strategies can be distinguished: time-dependent, resource-dependent and behaviour-dependent strategies (Faratin, Sierra, & Jennings, 1998). These strategies are well defined functions that may use various input parameters in order to produce the value of the issue under negotiation to be proposed at the current negotiation round. The proposed mechanism enhances any of the legacy strategies with learning techniques based on Neural Networks (NNs). In the studied framework, the NN-assisted strategies are used by the CA in order to estimate the future behaviour of the PA. This section presents the proposed NN-assisted strategy and describes the specifics of the NNs employed.

Enabling PA Behaviour Prediction

As already mentioned, the research presented in this paper aims to estimate the parameters governing the PA's strategy enabling the CA to predict the PA's future price offers. The objective is to decide at an early round whether to aim for an agreement with the specific PA, or withdraw from the negotiation thread as early as possible, if no agreement is achievable. For this purpose, two different Neural Networks (NNs) have been employed. These NNs are trained off-line with proper training sets and are then used during the on-line negotiation procedure whenever the CA requires so. The procedure starts normally, and as long as there are enough proposals made by the PA, the CA uses the NNs to make a reliable prediction of its opponent's strategy. This requires only a few negotiation rounds (compared to the CA's deadline expiration round) and

this is the main reason why this technique turns out to be significantly useful.

In addition to the $\left[p_m^a, p_M^a\right]$ interval, there are mainly 3 other parameters that determine the agent's negotiation strategy: parameter $k^a \in [0,1]$ that determines the initial offer made by the agent at $t = 0$, the concession rate $\beta > 0$ (Faratin, Sierra, & Jennings, 1998), and the PA's last round L_p. In this paper, k^a does not lie among the parameters for prediction as it is safely assumed that the PA initiates the procedure from its maximum price offer. Without loss of generality, we focus on the case where the PA follows a polynomial strategy of arbitrary concession rate and timeout.

The CA negotiates based on a legacy strategy until round L_C^d. Then, the CA makes use of the NNs to obtain estimations $\overline{\beta}$ and $\overline{L_p}$. Round L_C^d will be hereafter called the *prediction round*. In the experiments conducted we have $L_C^d = 30$ and $L_C = 100$. Based on the history of the PA's price offers, NNs attempt to produce a valid estimation of the PA's offer generation function. Then, the CA may determine whether the current negotiation thread can lead to an agreement or this is not feasible given the CA's deadline. Thus, the NN-assisted strategy enables the CA to save time and withdraw early from negotiation threads that will end unsuccessfully.

The Neural Networks Employed

In our framework, where the prediction of a continuous function is required, we selected to study two types of NNs having no feedback loops: the multilayer perceptron (MLP) NN and the Generalized Regression (GR) NN. The latter is a special case of a Radial Basis Function (RBF) NN that is more appropriate for on-line function approximation (Haykin, 1999). Both networks were selected because of their suitability in such kinds of problems (Haykin, 1999).

For the MLP, we used a training function based on the Levenberg-Marquardt algorithm (Hagan, Demuth, & Beale, 1996) as it is the most convenient for such problems. The network was properly trained over 190 different input vectors (200 epochs each) representing a different combination of PA's offers based on a specific strategy. The best MLP architecture was decided after extensive experiments and set to 23 (log-sigmoid) – 3 (linear) neurons, for the hidden and the output layer respectively. Similarly, the GR network was trained over 280 different vectors to achieve accurate performance characteristics resulting in a 280 (hidden RBF neurons) – 3 (output) architecture.

Both NNs are employed by CAs and can provide reliable prediction of the PA's behaviour, once sufficient input samples (proposals) are available. The experiments conducted and the NNs performance evaluation, are presented in the two following sections.

EXPERIMENTS

In this section, the experiments conducted to evaluate the performance of the designed MLP and the GR NNs concerning the estimation of the future behaviour of the negotiating PA are presented. The first experiments' family aims to compare the actual behaviour of the PA with the one predicted by the MLP and the GR NNs, when $\left[p_m^P, p_M^P\right] = [0,100]$, $L_p = 200$ and $\beta \in$ [0.1, 10]. The sample values for β are derived from a uniformly distributed random vector of 100 values in the aforementioned area: 50 $\beta < 1$ (Boulware) and 50 $\beta > 1$ (Conceder). The estimated parameters include: the future PA offers until the 100th negotiation round, the minimum PA price offer until then and the PA's concession rate (β). The second experiment family investigates the case where $\left[p_m^P, p_M^P\right] = [0,100]$, $\beta = 1$ and $L_p \in [150, 250]$. The sample values for L_p are: 150:1:250. The estimated parameters include: the future PA price offers until the 100th negotiation round and the minimum PA price offer until then.

As illustrated in Figure 1, where the first experiment set is depicted, the MLP- and the GR-NN perform very similarly, managing to accurately predict the PA's price offer in general. In the same Figure, one may observe that both NNs are used until $\beta \le 2.8$. For higher concession rates and for polynomial PA strategies, an agreement is reached before the 30th round and the NN is not necessary for opponent behaviour prediction. As depicted in Figure 2, where the NNs are tested over linear PA's strategy, the MLP- and the GR-NN perform almost identically estimating the PA's price offer with low error margin. However, the deviation between the actual and the estimated PA offers increases as the round index increases and the PA timeout decreases. This is due to the fact that both NNs have a tendency to slightly underestimate PA's concession rate, especially when $\beta \ge 0.5$. This is confirmed by Figure 3a, where the MLP-NN and the GR-NN estimations of the concession rate are depicted along with the actual β of the PA, over the entire set of conducted experiments. Finally, as depicted in Figure 3b, with regards to the

S

Figure 1. Actual PA price offer and PA price offer predicted by (a) a MLP-NN and (b) a GR-NN, for 100 negotiation rounds when $L_P = 200$, $p_m^P = 0$, $p_M^P = 100$ and $\beta \in [0.1, 10]$
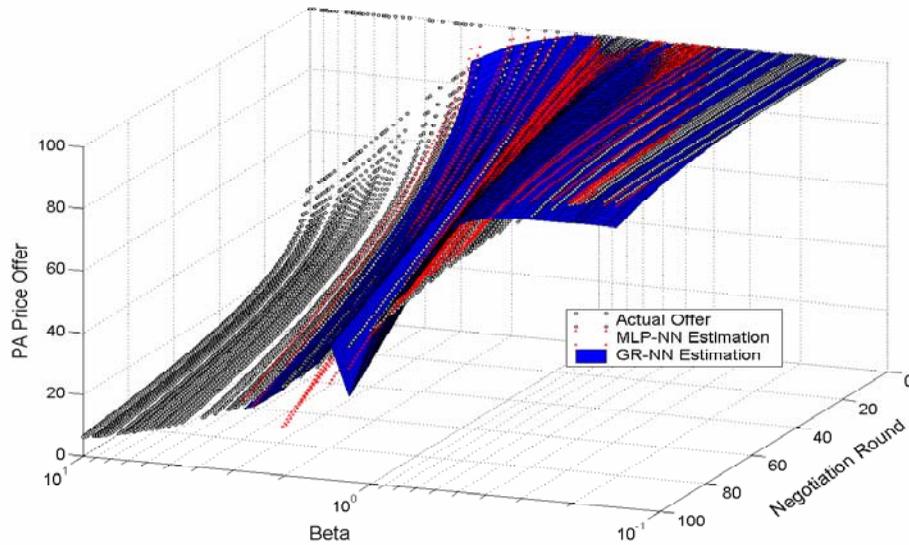
Figure 2. Actual PA price offer and PA price offer predicted by (a) an MLP-NN and (b) a GR-NN, for 100 negotiation rounds when $\beta = 1$, $p_m^P = 0$, $p_M^P = 100$ and $L_P \in [150, 250]$
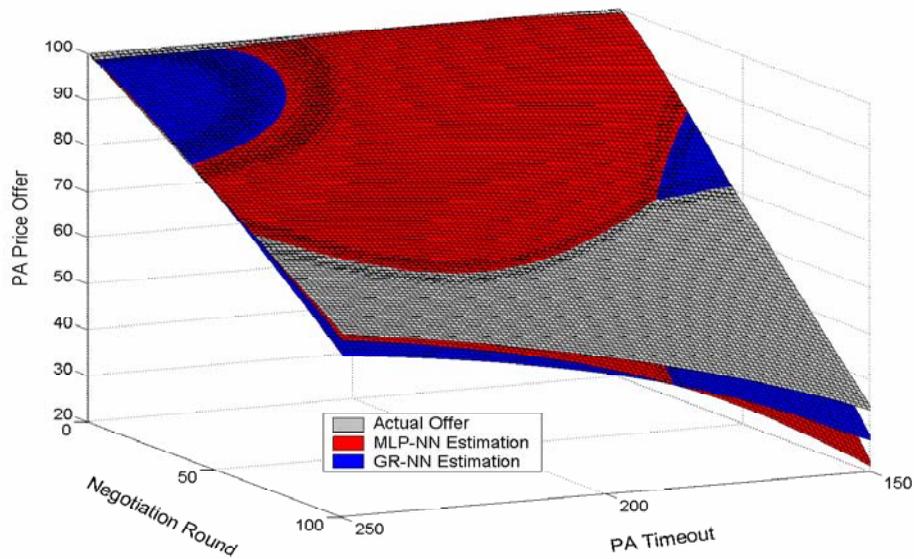
Figure 3. (a) Actual and estimated (by MLP and GR NNs) concession rate values when $L_P = 200$, $p_m^P = 0$, $p_M^P = 100$ and $\beta \in [0.1, 10]$. (b) Actual and estimated (by MLP and GR NNs) PA minimum price offer for all the experiments conducted in both families.

(a)

(b)
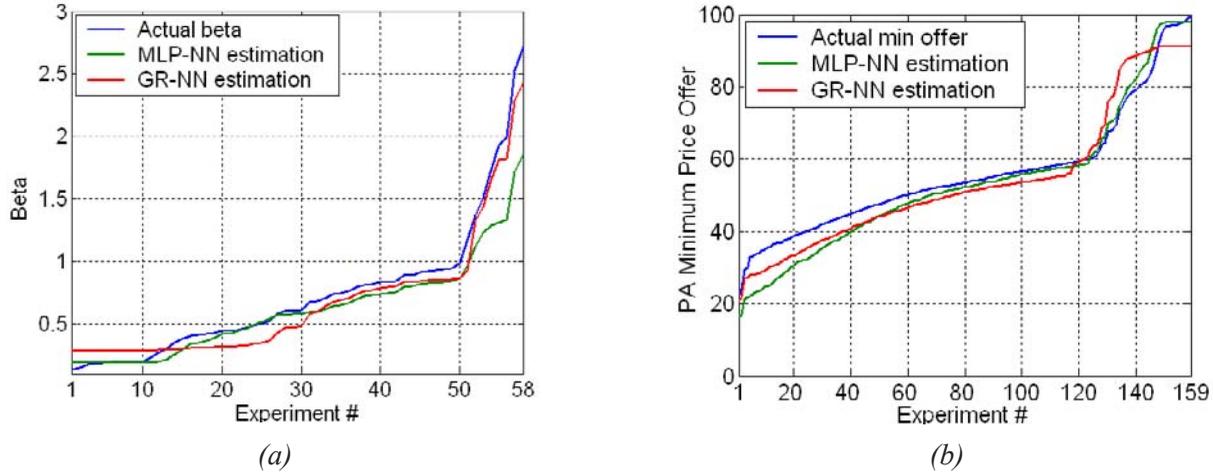
estimation of the PA's minimum price offer, the MLP slightly outperforms the GR. A brief analysis of all these findings is presented in the subsequent section.

EVALUATION

In Table 1 comparative results for two experiment families are illustrated with regards to the mean estimation errors of the MLP and the GR NNs concerning the PA price offer, the PA minimum price offer and the PA's concession rate. For all experiment families we have $[p_m^P, p_M^P] = [0,100]$. The rest of the parameter settings are presented in the table's first column, while at the second column the number of the experiments where the NN estimation was used is depicted. The results presented in the rest of the table indicate that the MLP NN slightly outperforms the GR NN with regards to the PA (minimum) price offer estimation demonstrating 0.5% - 2.4% higher accuracy in average. However, the opposite stands concerning the PA beta estimation, as the GR NN provides more accurate estimations by more than 3% in average.

In Table 2, evaluation results for the two NN-assisted negotiation strategies are illustrated for both experiment families assuming that $p_M^C = 50$[1]. These results include the number unsuccessful negotiation threads (UNTs) (that are due to the fact that $p_m^P > p_M^C$)

) and the duration of the UNTs ($L_C = 100$[2]) in case no opponent behaviour prediction mechanism is used, the number of UNTs detected by the NNs at round 30, the UNTs that were detected and thus terminated early by the NNs, the mean duration of the UNTs and the mean UNT duration decrease. These results indicate that the MLP and the GR NNs manage to identify ~91% and ~83% of the UNTs in average, respectively. Furthermore, the MLP and the GR NNs achieve ~64% and ~58% reduction of the UNTs' duration in average, respectively. With regards to the elimination of the UNTs, the MLP-assisted strategy clearly outperforms the GR-assisted negotiation strategy. For the reasons above, it is estimated that MLP NNs are more appropriate for assisting negotiating intelligent agents to predict their opponent's behaviour at an early negotiation round in case the agent values a timely detection of unsuccessful negotiation threads.

CONCLUSION

This paper proposed to use Neural Networks in order to enhance negotiating agents with learning techniques enabling them to predict their opponents' negotiation behaviour. The designed NN-assisted negotiation strategy turns out to be very useful, as it leads to substantial duration reduction of unsuccessful negotiation threads,

Table 1. Comparative results concerning the mean estimation error of the two NN-assisted negotiation strategies for the PA price offers, for the PA min offer and the PA concession rate.

Experiment Settings	Times NN-estimation was used	Mean [price-offer estimation error]		Mean [min-price-offer estim. error]		Mean [beta estimation error]	
		MLP	GR	MLP	GR	MLP	GR
$\beta \in [0.1, 10]$, $L_P = 200$	4118	0.97%	2.12%	0.41%	2.80%	15.65%	8.26%
$L_P \in [150, 250]$, $\beta = 1$	7171	1.21%	1.71%	8.26%	8.91%	12.51%	12.73%
OVERALL	11289	1.12%	1.86%	5.40%	6.68%	13.92%	10.72%

Table 2. Comparative results concerning the unsuccessful negotiation thread detection by the two NN-assisted negotiation strategies

Experiment Settings	# Unsuc. Negot. Threads (UNTs)	Mean duration of UNTs (no NN)	# UNTs detected at round 30		UNTs' elimination ratio		Mean UNTs' duration		Mean UNTs' duration decrease	
			MLP	GR	MLP	GR	MLP	GR	MLP	GR
$\beta \in [0.1, 10]$, $L_P = 200$, $p_M^C = 50$	50	100	49	49	98.0%	98.0%	31.4	31.4	68.6%	68.6%
$L_P \in [150, 250]$, $\beta = 1$, $p_M^C = 50$	51	100	43	35	84.3%	68.6%	41.0	52.0	59.0%	48.0%
OVERALL	101	100	92	84	91.1%	83.2%	36.2	41.8	63.8%	58.2%

due to the fact that the cases where agreements are not achievable are detected at an early stage. Thus, the NNs support the decision of the agents to withdraw or not from the ongoing negotiation threads. More specifically, when the CA uses the NN-assisted strategies it is capable of predicting its opponent's behaviour with significant accuracy, thus getting aware of the potential outcome of the negotiation. Both the MLP and the GR NNs studied demonstrate average opponent price offer estimation error lower than 2% and PA min acceptable price estimation error ~6%. Additionally, the unsuccessful negotiations are detected by the MLP NN in more than 90% of the cases in average, demonstrating ~8% better overall performance than the GR NN. Thus, the MLP NN is proven to be more appropriate, when the CA aims to avoid a possible unprofitable or even unachievable agreement. This leads to minimization of the required time and processing resources and to maximization of the CAs overall profit from a series of threads for a single commodity.

REFERENCES

Buttner, R. (2006). A Classification Structure for Automated Negotiations. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (IAT 2006), Hong Kong, China.

Faratin, P., Sierra, C., & Jennings, N. (1998). Negotiation Decision Functions for Autonomous Agents. *International Journal of Robotics and Autonomous Systems*. 24(3-4), 159-182.

Fatima, S., Wooldridge, M., & Jennings, N. (2007). On Efficient Procedures for Multi-issue Negotiation. In Agent-Mediated Electronic Commerce: Automated Negotiation and Strategy Design for Electronic Markets. *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg New York. 4452, 31-45.

Hagan, M., Demuth, H., & Beale, M. (1996). *Neural Network Design*. Boston MA USA: PWS Publishing Company.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd edition). London UK: Prentice Hall.

Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., & Wooldridge, M. (2001). Automated negotiation: Prospects, methods and challenges, *International Journal of Group Decision and Negotiation*, (10)2, 199-210.

Kraus, S. (2001). *Strategic Negotiation in Multiagent Environments*. MA USA: MIT Press.

Li, H., Su, S., & Lam, H. (2006). On Automated e-Business Negotiations: Goal, Policy, Strategy, and Plans of Decision and Action. *Journal of Organizational Computing and Electronic Commerce*. (16)1, 1-29

Maes, P., Guttman, R., & Moukas, A. (1999). Agents that Buy and Sell: Transforming Commerce as we Know It. *Communications of the ACM*, 42(3), 81–91.

Muller, H. (1996). Negotiation principles. In: *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons New York USA. 211-229.

Rosenschein, J., & Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MA USA: MIT Press.

Saha, S. (2006). *Negotiating contracts in multiagent societies*. Doctoral Thesis, University of Tulsa, USA.

Sierra, C. (2004). Agent-Mediated Electronic Commerce. *Journal of Autonomous Agents and Multi-Agent Systems*, (9)3, 285-301.

Silva, A., Romão, A., Deugo, D., & da Silva, M. (2001). Towards a Reference Model for Surveying Mobile Agent Systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(3), 187-231.

Walton, D., & Krabbe, E. (1995). *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. Albany NY, USA: SUNY Press.

KEY TERMS

Automated Negotiation: It is the process by which group of actors communicate with one another aiming to reach to a mutually acceptable agreement on some matter, where at least one of the actors is an autonomous software agent.

Bilateral Negotiation: A negotiation procedure, where exactly two parties are involved, i.e. a client and a provider.

Backpropagation Algorithm: A supervised learning technique used for training artificial NNs based on the minimisation of the error obtained from the comparison between the desired output and the actual one when applying specific inputs.

Generalized-Regression (GR) NN: A GR NN is a special case of a RBF NN with a second linear layer and is often used for function approximation.

Multi-Layer Perceptron (MLP): A fully connected feedforward NN with at least one hidden layer that is trained using back-propagation algorithmic techniques.

Negotiation Protocol: The set of rules that govern the interactions between the negotiating parties.

Negotiation Strategy: The reasoning model based on which the negotiating parties formulate their response to their opponent's offers.

Neural Network (NN): A network modelled after the neurons in a biological nervous system with multiple synapses and layers. It is designed as an interconnected system of processing elements organized in a layered parallel architecture. These elements are called neurons and have a limited number of inputs and outputs. NNs can be trained to find nonlinear relationships in data, enabling specific input sets to lead to given target outputs.

Radial Basis Function (RBF): Function that involves a distance criterion with respect to a centre, such as a circle, ellipse or Gaussian.

RBF NN: It is an artificial NN, the activation functions of which are radial basis functions. It has two layers of processing, where the first maps the input onto each RBF neuron in the other (hidden) layer.

ENDNOTES

[1]   We selected the p^C_M to be equal to the median value in the PA's acceptable price interval.

[2]   To be more accurate, the duration of UNTs is equal to: $\min(L_C, L_P)$. However, in this paper's study, we always have $L_C < L_P$, and thus the duration of UNTs is equal to L_C.

Signed Formulae as a New Update Process

Fernando Zacarías Flores
Benemérita Universidad Autónoma de Puebla, Mexico

Dionicio Zacarías Flores
Benemérita Universidad Autónoma de Puebla, Mexico

Rosalba Cuapa Canto
Benemérita Universidad Autónoma de Puebla, Mexico

Luis Miguel Guzmán Muñoz
Benemérita Universidad Autónoma de Puebla, Mexico

INTRODUCTION

The **agent** paradigm has recently increased its influence in the research and development of computational logic-based systems. A clear and correct specification is made through Logic Programming (LP) and **Non-nomotonic Reasoning** that have been brought (back) to the spotlight. Also, the recent significant improvements in the efficiency of *LP* implementations for Non-monotonic Reasoning (De Schreye, Hermenegildo & Pereira, 1999) have helped to this resurgence. However, the agents need update constantly their knowledge base and, particularly the intentional base (rules) such that our agent has the ability to reacting to changes in dynamic environments is of crucial importance within the context of software agents. Such feature should correspond to a deliberative rational behavior wanted for our agents.

The quality of the service that an agent offers is based on the form in which an agent combines rationality and reactivity. A reactive agent can offer well evaluated recommendations but, this response is based on outdated information, while a rational behavior may generate recommendations based on the most recently acquired information. So, we are interested in developing environment-aware agents. For this reason, is very important to have an update process for agents, i.e., that it allows us to design agents with its rational component.

Over recent years, several semantics for logic program **updates** have been proposed (Brewka, Dix, & Knonolige 1997) (De Schreye, Hermenegildo, & Pereira, 1999) (Katsumo & Mendelzon, 1991). All these semantic ones coincide in considering the AGM proposal as the standard model in the update theory, for their wealth in **properties**. The AGM approach, introduced in (Alchourron, Gardenfors & Makinson, 1985) is the dominating paradigm in the area, but in the context of monotonic logic. All these proposals analyze and reinterpret the AGM postulates under the **Answer Set Programming** (ASP) such as (Eiter, Fink, Sabattini & Thompits, 2000). However, the majority of the adapted AGM and update postulates are violated by update programs, as shown in (De Schreye, Hermenegildo, & Pereira, 1999). For this reason, we have been working in finding properties that our update operator satisfies (Osorio & Zacarías, 2003) (Zacarías & Osorio, 2005) (Arrazola & Zacarias, 2005). Our purpose is to build a semantics based on structural properties. This is our main objective in the **update** theory. In (De Schreye, Hermenegildo, & Pereira, 1999) (Osorio & Zacarias, 2003) (Zacarías, Osorio & Arrazola, 2005) (Zacarias, 2005) the authors present a set of properties that the update operator satisfies. In this paper we continue with this same research line presenting a novel proposal with the aim to enrich the **update theory** that we have begun in (Osorio & Zacarias, 2003) (Zacarías, Osorio & Arrazola, 2005) (Zacarias, 2005). This novel proposal contributes with two benefits. First, we conserve many of the properties presented in previous works (Osorio & Zacarias, 2003) (Zacarías, Osorio & Arrazola, 2005) (Zacarias, 2005), such as: Weak Irrelevance of Syntax (WIS). This property is similar to one postulate proposed by AGM, but in this case for **nonmonotonic logic** and under **Answer Set Programming** (ASP) introduced and defined by (Gelfond & Lifschitz, 1988).

BACKGROUND

In this section, we present advances in the updates context. Also, we give some general definitions for our theory. We define our theory about logic programs.

Advances on Updates

We consider the task of updating logic programs under **non-monotonic reasoning** and a purely logical view. Since an intelligent agent is situated in an environment which is subject to change, it is required the agent to be adapted over time. For agents utilizing logic programming techniques for representing their knowledge, it is required the agent to be capable of updating logic programs accordingly, in order to ensure adaptability. We chose one of the approaches; viz. update answer set semantics (Zacarías, 2005) (Osorio & Zacarías, 2003) (Eiter, Fink, Sabattini & Thompits, 2000) (Banti, Alferes & Brogi, 2003). Resides, an underlying update semantics, which specifies how new, possibly inconsistent information, have to be incorporated into the knowledge base, an agent needs to have a certain **update** policy, i.e., a specification of how to react upon the arrival of an update. The issue of how to specify change requests for knowledge bases has received growing attention more recently and suitable specification languages for **non-monotonic logic** programs have been developed (Leite, 2001) (Leite, 2002).

In (Zacarías, 2005) we have introduced a new proposal towards the enrichment of the update operator "⊕". There, we have presented a refinement of the **stable model** semantics for the update operator. Also, we presented a new property that allows us to face updates where new information contains rules that define a conservative extension. So, we gave an extension of our properties proven in (Osorio & Zacarías, 2003), under N logic. This approach is based on the work made by Eiter et al. (Eiter, Fink, Sabattini & Thompits, 2000), and inspired in a recent approach presented by Alferes et al. (Banti, Alferes & Brogi, 2003). With this work, we improve and enrich the update operator proposed by Eiter et al. (Eiter, Fink, Sabattini & Thompits, 2000), giving as result a new update operator.

UPDATES FOR REAL TIME APPLICATIONS

In this section we present a novel mechanism that allows updating a knowledge base in a quick and easy way. Furthermore, this proposal satisfies similar structural properties to those that we have presented in previous works. So, we give the basic concepts for our theory and we present our main contribution based on **signed formulae** (Ariely, Denecker, Nuffelen & Bruynooghe, 2004).

Preliminary

Rules are built from propositional atoms and the 0-place connectives T and ⊥ using negation as failure (¬) and conjunction (,). A *rule* is an expression of the form:

$$\text{Head} \leftarrow \text{Body} \tag{1}$$

If Body is T then we identify rule (1) with rule Head. If a Head is ⊥ then we identify rule (1) with a restriction. A *program* is a set of rules. A logic program P is a (possibly infinite) set of rules. For a program P, I is a model of P, denoted $I \models P$, if $I \models L$ for all $L \in P$. As it is shown in (Brewka, Dix, & Knonolige, 1997), the Gelfond-Lifschitz transformation for a program P and a model $N \subseteq B_P$ (B_P denotes a set of atoms that appear in P) is defined by

$$P^N = \{rule^N : rule \in P\}$$

where $(A \leftarrow B_1, \ldots, B_m, \neg C_1, \ldots, \neg C_n)^N$ is either:

a. $A \leftarrow B_1, \ldots, B_m$, if $\forall j \leq n: Cj \notin N$;
b. T,

otherwise

Note that P^N is always a definite program. We can therefore compute its least Herbrand model (denoted as M_{PN}) and check whether it coincides with the model N which we started with:

Definition 1. (Gelfond & Lifschitz, 1988) N is a **stable model** of P iff N is the minimal model of P^N

DATABASE REPAIRS

In this document we present our more recent approach about update under repairs to database. The way in that we approach the update problem is based on (Ariely, Denecker, Nuffelen & Bruynooghe, 2004), i.e., based on the idea of **repair inconsistent database**. So, given a possibly inconsistent database this mechanism represents the possible ways to restore its consistency in terms of **signed formulae**. In our context, this mechanism is used to incorporate new information that can be inconsistent with the previous database containing the **agent's knowledge**.

In a similar form as in (Ariely, Denecker, Nuffelen & Bruynooghe, 2004), L defines a first order language, S is a fixed database schema and D a fixed domain. A database instance D consists of atoms in the language L, where D has a finite active domain, $A(D)$, which is a subset of D.

Definition 2. (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) a database is a pair (*D, IC*), where *D* is a database instance, and *IC*, the set of integrity constraints, is a finite and classically consistent set of formulae in L.

So, a database is (Ariely, Denecker, Nuffelen & Bruynooghe, 2004): $DB = (D, IC)$, let

$$DB^A = D \cup ICA = D \cup \{\rho(\psi) \mid \psi \in IC, \rho : \text{var}(\psi) \to A(D)\}$$

Where ρ is a ground substitution of variables to the individuals of $A(D)$, the active domain of *D*, *DB* is called the Herbrand expansion of *DB*. As *D*, *IC* and $A(D)$ are all finite sets, DB^A is also finite, and so $\Sigma^{DB} = \{p_1, p_2, \ldots, p_n\}$, the set of the atomic formulae that appear in DB^A, is finite as well.

UPDATES BASED ON SIGNED FORMULAE

For their simplicity our new **update** process is suitable for applications that have to give answers in real time. In (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) the authors suppose that the database is inconsistent, and then they give a general description of how to restore the consistency of databases instances that do not satisfy a given set of integrity constraints. Here, we adapt this method to updates of logic programs and we illustrate that this can be used efficiently in our context.

Here, we consider updates in the setting of logic programs, i.e., we consider that a database is represented by a logic program. So, in our context, we start with the fact that our database is updated with new information that it causes an **inconsistent database**. At this moment, we apply this method with the objective of making our database consistent. Follow, we present a general framework used in (Ariely, Denecker, Nuffelen & Bruynooghe, 2004). A database is a pair (D, IC), where D is a database instance, and IC, the set of integrity constraints, is a finite and classically consistent set of formulae in a language.

Given a possibly inconsistent database, our goal is to restore its consistency, i.e., to repair the database:

Definition 3. Similarly as in (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) an update of a database *DB = (D, IC)* is a pair {**Insert, Retract**}, s.t. **Insert** ∩ D = ϕ and **Retract** $\subseteq$ D. A pair of DB is an update of DB, for which (*D $\cup$ Insert \ Retract, IC*) is a consistent database.

The intuitive meaning is as follows: a database is **updated** by inserting the elements of **insert** and removing the elements of **Retract**. An update is a pair when the resulting database is consistent. Note that if *DB* is consistent, then (ϕ, ϕ) is a pair of DB.

As follows, we give some examples that illustrate how this mechanism is adapted to updates.

Example 1. This example illustrates a daily update regarding the energy flaw (Eiter, Fink, Sabattini & Thompits, 2000). Suppose that you have the following database:

DB: sleep ← ¬tv-on.
    night.
    tv-on.
    watch-tv ← tv-on.
    ← power-failure, tv-on.

Here, the DB is consistent, however, if we update the DB with the following rule:

power-failure.

This DB is not consistent. Following the format defined in (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) this example is the pair:

DB = ({sleep ← ¬tv-on., night., tv-on., watch-tv ← tv-on., power-failure.}, {← power-failure, tv-on.})

Therefore, we can adapt the method proposed in (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) and to carry out a correct update of this database as follows:

First, we add our initial configuration the new information (update):

DB = ({sleep ← ¬tv-on., night., tv-on., watch-tv ← tv-on., power-failure.}, {←power-failure, tv-on, power-failure.})

Second, we obtain the **signed formula**:

(¬power-failure ∨ ¬tv-on) ∧ power-failure.

Using format of (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) we obtain:

$(\neg\neg S_{power\text{-}failure} \vee \neg\neg S_{tv\text{-}on}) \wedge \neg S_{power\text{-}failure}.$

it is equivalent to:

$(S_{power\text{-}failure} \vee S_{\text{---}tv\text{-}on}) \wedge \neg S_{power\text{-}failure}.$

Third, we calculate v^R the valuation that is associated with R, obtaining:

$v^R(S_{power\text{-}failure}) = 1$
$v^R(S_{tv\text{-}on}) = 1$
$v^R(\neg S_{power\text{-}failure}) = 0$

Therefore, we have: ({power-failure}, {tv-on})
This means that the suggested update is:

Insert → {power-failure}
Retract → {tv-on}

As we can see, this is the wanted result.
Now, we show another example using the method presented in (Eiter, Fink, Sabattini & Thompits, 2000).

Example 2. This program describes some knowledge about the sky (Banti, Alferes & Brogi, 2003). Suppose that you have the following database:

DB: day ← ¬night.
  ¬see-stars.
  ← see-stars, day.

Here, the DB is consistent, however, if we update the DB with the following rule:

see-stars.

This DB is not consistent. Following the format defined in (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) this example is the pair:

DB = ({day ← ¬night., ¬see-stars.}, {←see-stars, day.})

Therefore, we can adapt the method proposed in (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) obtaining a correct update of this database as follows:

First, we add our initial configuration to the new information (update):

DB = ({day ← ¬night., ¬see-stars.}, {← see-stars, day, see-stars.})

Second, we obtain the **signed formula**:

(¬see-stars ∨ ¬day) ∧ see-stars.

Using format of (Ariely, Denecker, Nuffelen & Bruynooghe, 2004) we obtain:

$(\neg\neg S_{see\text{-}stars} \vee \neg\neg S_{day}) \wedge \neg\neg S_{see\text{-}stars}.$

it is equivalent to:

$(S_{see\text{-}stars} \vee S_{day}) \wedge S_{see\text{-}stars}.$

Third, we calculate v^R the valuation that is associated with R, obtaining:

$v^R(S_{see\text{-}stars}) = 1$
$v^R(S_{day}) = 1$
$v^R(S_{see\text{-}stars}) = 1$

Therefore, we have: ({see-stars}, {day})
This means that the suggested update is:

Insert → {see-stars}
Retract → {day}   this is, as desired.

FUTURE TRENDS

Just as in (Eiter, Fink, Sabattini & Thompits, 2000) we coincide that because of apparent lack of minimality of change, we then considered refinements of the semantics in terms of minimal and strictly minimal answer sets. Several issues remain for further work. An interesting point (Eiter, Fink, Sabattini & Thompits, 2000) concerns the formulation of postulates (principles or properties) for update operator on logic programs and, more generally, on **non-monotonic theories**. As you can see in (Eiter, Fink, Sabattini & Thompits, 2000), several postulates from the area of logical theory change fail for update programs. This may be explained by the dominant role of syntax for update embodied by causal rejection of rules.

We have extended our operator to Pstable semantics. However, we should continue working in this line, since there are properties such as the following:

$$P \oplus P_1 \oplus Q \equiv P \oplus P_2 \oplus Q$$

for every program P_i and Q, that are not always satisfied, due to our property is just satisfied by the right.

We consider that the future researches rotated around the search of properties that the update operator satisfies, and that in principle they have been begun by some authors (Osorio & Zacarías, 2003) (Banti, Alferes & Brogi, 2003) (Eiter, Fink, Sabattini & Thompits, 2000) (Zacarías, Osorio & Arrazola, 2005) (Zacarías, 2005).

CONCLUSION

In this paper, we considered a new proposal to provide an update process to our agents. Our proposal is a novel and simple methodology that allows an agent to maintain updated its knowledge base in all moment. This provides an agent to behave in a rational way, similar to human behavior. Furthermore, it is an appropriate proposal for applications that require answers in real time. Also, this proposal opens the possibilities for building real-life applications, like intelligent agents whose rational component is modelled by a knowledge base, which is in turn maintained using **update logic programs**.

REFERENCES

Alchourron C.E., Gardenfors P., & Makinson D. On the logic of Theory Change, Partial Meet Functions for Contraction and Revision Functions. Journal of Symbolic Logic, 50:510-530, 1985.

Ariely O., Denecker M., Van Nuffelen & Bruynooghe M.. Database repair by Signed formulae In D. Seipel and J.M. Turrul, editors, Foundations of Information and Knowledge Systems, Third International symposium, FoIKS 2004, Wilhelminenburg Castle, Austria, vol. 2942 LNCS, pp. 231–241, 2004.

Arrazola J., Dix J., & Osorio M. Confluent term rewriting systems for non-monotonic reasoning. Computacion y Sistemas, II(2-3):299–324, 1999.

Banti F., Alferes J. & Brogi A. A principled semantics for logic program updates. In M. Gelfond, N. Leone and P. Pfeifer, editors, proceedings into Eighteenth International Join Conference, LNAI, México, Springer Verlag, 2003.

Brewka G., Dix J., & Knonolige K. Nonmonotonic reasoning, an overview. CSLI Publication Eds. Leland Stanford Junior University, 1997.

De Schreye D., Hermenegildo M. & Pereira L.M. Paving the Roadmaps: Enabling and Integration Technologics, 1999.

http://www.compulog.org/net/Forum/Supportdocs.html

Eiter T., Fink M., Sabattini G., & Thompits H. Considerations on Updates of Logic Programs. In M.O. Aciego, LP. de Guzmßn, G. Brewka, and L.M. Pereira, editors, Proc. Seventh European Workshop on Logic in Artificial Intelligence JELIA 2000, vol. 1919 in Lecture Notes in Artificial Intelligence. LNAI, Springer 2000.

Gelfond M., & Lifschitz V. The stable model semantics for logic programs. Proceedings of the Fifth International Conference on Logic Programming 2, MIT Press. Cambridge, Ma. pp.1070-1080, 1988.

Kahneman D. & Tversky A. Subjective probability: A judgment of representativeness. Cognitive Psychology, 3:430–454, 1972.

Kahneman D. & Tversky A. On the psychology of prediction. Psychological Review, 80:237–251, 1973.

Katsumo H. & Mendelzon A.O. On the difference between updating a knowledge base and revising it. in: J.A. Allen, R. Fikes and E. Sandewell. eds.. Principles of knowledge representation and reasoning: Proceedings of the Second International Conference (Morgan Kautmann. San Mateo. CA. 1991) pp. 387--394.

Keller A. & Winslett M. On the use of an extended relational model to handle changing incomplete information. IEEE Transactions on software engineering, 11(7):620-633, 1985.

Leite J.A. A modified semantics for lups. In Proceedings of the 10th Protuguese Conference on Artificial Intelligence (EPIA 2001), pages 261–275. Lecture Notes on Computer Sciences, 2001.

Leite J.A. Evolving Knowledge Bases – Specification and Semantics. PhD thesis, Departamento de Inform´atica, Universidade Nova de Lisboa, 2829-526, 2002.

Lifschitz V., Pearce D., & Valverde A. Strongly equivalent logic programs. ACM Transactions on Computational Logic, 2:526-541, 2001.

Osorio M, Navarro J.A., & Arrazola J. Equivalence in answer set programming. In A. Pettorossi, editor, Logic Based Program Synthesis and Transformation. 11th International Workshop, LOPSTR 2001, number 2372 in LNCS, pages 57–75, Paphos, Cyprus, November 2001. Springer.

Osorio M., Navarro J.A., & Arrazola J. Answer set programming and S4. Submitted to publication, 2002.

Osorio M., Navarro J.A., & Arrazola J.. Applications of intuitionistic logic in answer set programming (extended version). TPLP, 1, 2003.

Osorio M. & Zacarias F., Irrelevance of Syntax in updating answer set programs, Proceedings Of Fourth Mexican International Conference On Computer Science Enc'03, pp.183-188, Eds. J. H. Sossa, and E. Perez, México, 2003.

Pearce D. From Here to There: Stable negation in Logic Programming, in D. Gabbay, H. Wansing (Eds.) What is Negation? Kluwer Academic Publishers, Dordrecht.

Tversky A. & Kahneman D. Belief in the law of small numbers. Psychological bulletin, 2:105–110, 1971.

Tversky A. & Kahneman D. Judgment under uncertainly: Heuristics and biases. American association for Advancement of Science, 1:1124–1131, 1971.

Uceda F., Zacarías F. & Zacarías D. Pstable tool and its application, to appear in journal of Engineering Letters, special issue on Computer Science and Artificial Intelligence, Spain 2007.

Zacarías F., Osorio M., & Arrazola J. Updates based on Structural Properties –USP-. Gests international transactions on computer science and engineering, pp. 61-72, issn: 1738-6438, isbn: 89-953729-5-8, October 2005.

Zacarías F. Belief Revision and Updates in Commonsense Reasoning, Ph. D thesis, Universidad de las Américas Puebla, 2005.

Zacarías F, Osorio M. & Fernández E. Updates under Pstable, to appear in journal of Engineering Letters, special issue on Computer Science and Artificial Intelligence, Spain 2007.

Zacarías F. & Téllez A. Programación lógico–funcional. In CONIELECOMP 2002, pages 45–49, Acapulco (México), 2002.

KEY TERMS

Beliefs: An agent whose knowledge base is the theory *T* believes *F* if and only if *F* belongs to every intuitionistically complete and consistent extension of *T* by adding only negated literals.

Equivalence: Two programs are *equivalent* if they have exactly the same answer sets.

Intelligent Agent: An intelligent agent is a component of software (or hardware) that it perceives and it acts autonomously in an open and dynamic environment, learning and cooperating with other agents (the same user) to offer a benefit to their user.

Principle of Irrelevance of Syntax: The meaning of the knowledge that results from an update must be independent of the syntax of the original knowledge, as well as independent of the syntax of the update itself.

Strong Equivalence: (Lifschitz, Pearce & Valverde, 2001). We say that P_1 and P_2 are *strongly equivalent* if for every program P, $P_1 \cup P$ and $P_2 \cup P$ have the same answer sets.

Update: Let P be the program representing the current knowledge base, if it is updated by another program U, then P_U is a program updated of P if only if the models of P_U are the result of updating each of the models of P according to a given semantics S; to each of these models apply the update request U to obtain a new set of models *M*; P_U is any logic program whose models are exactly *M*.

Weak Irrelevance of Syntax: $T_1 \equiv T_2$ implies *Bel(K ∇ T_1) = Bel(K ∇ T_2),* where *K*, T_1 and T_2 are any theories, Bel(*T*) defines the set of answer sets of *T*, ∇ is the update operator, and understanding that equivalence means that both programs (T_1 and T_2) have the same answer sets.

Solar Radiation Forecasting Model

Fatih Onur Hocaoğlu
Anadolu University Eskisehir, Turkey

Ömer Nezih Gerek
Anadolu University Eskisehir, Turkey

Mehmet Kurban
Anadolu University Eskisehir, Turkey

INTRODUCTION

The prediction of hourly solar radiation data has important consequences in many solar applications (Markvart, Fragaki & Ross, 2006). Such data can be regarded as a time series and its prediction depends on accurate modeling of the stochastic process. The computation of the conditional expectation, which is in general non-linear, requires the knowledge of the high order distribution of the samples. Using a finite data, such distributions can only be estimated or fit into a pre-set stochastic model. Methods like Auto-Regressive (AR) prediction, Fourier Analysis (Dorvlo, 2000) Markov chains (Jain & Lungu, 2002) (Muselli, Poggi, Notton & Louche, 2001) and ARMA model (Mellit, Benghanem, Hadj Arab, & Guessoum, 2005) for designing the non-linear signal predictors are examples to this approach. The neural network (NN) approach also provides a good to the problem by utilizing the inherent adaptive nature (Elminir, Azzam, Younes, 2007). Since NNs can be trained to predict results from examples, they are able to deal with non-linear problems. Once the training is complete, the predictor can be set to a fixed value for further prediction at high speed. A number of researchers have worked on prediction of global solar radiation data (Kaplanis, 2006) (Bulut & Buyukalaca, 2007). In these works, the data is treated in its raw form as a 1-D time series, therefore the inter-day dependencies are not exploited. This article introduces a new and simple approach for hourly solar radiation forecasting. First, the data are rendered in a matrix to form a 2-D image-like model. As a first attempt to test the 2-D model efficiency, optimal linear image prediction filters (Gonzalez, 2002) are constructed. In order to take into account the adaptive nature for complex and non-stationary time series, NNs are also applied to the forecasting problem and results are discussed.

BACKGROUND

This article presents a two-dimensional model approach for the prediction of hourly solar radiation. Before proceeding with the prediction results, the following technical background is provided. Using the described tools, the approach is tested with optimal coefficient linear filters and artificial NNs (Hocaoglu, Gerek & Kurban, 2007).

The 2-D Representation of Solar Radiation Data

The collected hourly solar radiation data is a 1-D discrete-time signal. In this work, we render this data in a 2-D matrix form as given in equation 1.

$$Rad = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

(1)

where the rows and columns of the hourly solar radiation matrix indicate days and hours, respectively. Such 2-D representation provides significant insight about the radiation pattern with time. First surface plot of the data is obtained then image view of the data is obtained and given in Fig 1.

By inspecting the image version of the data in Fig. 1, it is easy to interpret daily and seasonal behavior of solar radiation. Dark regions of the image indicate that there is no sun shine on horizontal surface. The transition from black to white indicates that solar radiation fall on horizontal surface is increasing or decreasing. During winter time, the dawn to dusk period is shorter, producing a narrower protruding blob. Conversely, the

Figure 1. Image view of solar radiation data
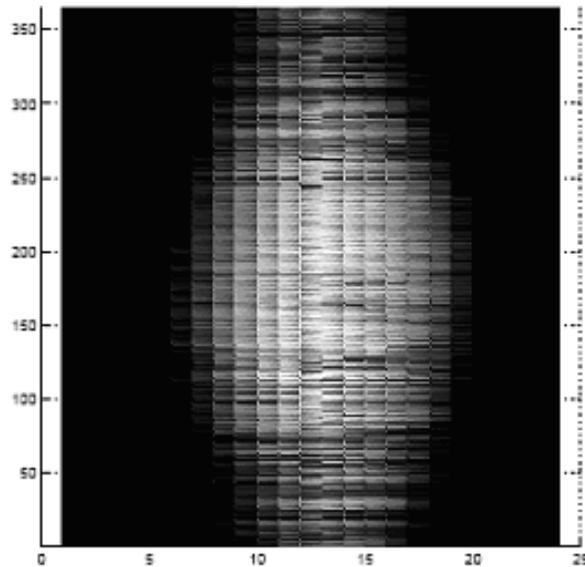
white blob is wider during summer times, indicating that the day-time is longer. The width behavior of the white blob clearly indicates the seasonal changes of sun-light periods. The horizontal *and* vertical correlations within the 2-D data are quite pronounced. This implies that, given the vertical correlation among the same hours of consecutive days, it is beneficial to use 2-D prediction for hourly forecasting. The prediction efficiency of the proposed model is illustrated with 2-D optimum linear prediction filters and NNs.

Optimal 2-D Linear Prediction Filter Design

Due to predictive image coding literature, it is known that a 2-D matrix can be efficiently modeled by linear predictive filters (Gonzales, 2002) (Sayood, 2000). The prediction domain is a free parameter determined according to the application. Consider a three coefficient prediction filter structure as given in expression 2:

C	$x_{i,j+1}$
$x_{i+1,j}$	$\hat{x}_{i+1,j+1} = ?$

(2)

The linear filter coefficients a_1, a_2 and a_3 are optimized, and the prediction result $\hat{x}_{i+1,j+1}$ is estimated as

$$\hat{x}_{i+1,j+1} = x_{ij}.a_1 + x_{i(j+1)}.a_2 + x_{(i+1)j}.a_3 \qquad (3)$$

The prediction error for this term is:

$$\varepsilon_{i+1,j+1} = \hat{x}_{i+1,j+1} - x_{i+1,j+1} \qquad (4)$$

The total error energy corresponding to the whole image prediction can be calculated as:

$$\varepsilon = \sum_{i=2}^{m} \sum_{j=2}^{n} \varepsilon_{ij}^2 \qquad (5)$$

where *m* and *n* correspond to the width and height of the image, which are, for the solar data, 365 and 24, respectively. The filter coefficients that minimize this function can be found from the solution of the minimization derivative equation:

$$\frac{\partial \varepsilon}{\partial a_1} = \frac{\partial \varepsilon}{\partial a_2} = \frac{\partial \varepsilon}{\partial a_3} = 0 \qquad (6)$$

The solution to equation 6 yields the following matrix-vector equation:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{7}$$

which is compactly written as $\mathbf{R} \cdot \mathbf{a} = \mathbf{r}$, so the optimal filter coefficients can be obtained as

$$\mathbf{a} = \mathbf{R}^{-1}\mathbf{r} \tag{8}$$

A Brief Discussion on Learning Techniques of NNs

There are several techniques to achieve high speed NN algorithms. Among these techniques, heuristic techniques were developed from an analysis of the performance of the standard steepest descent algorithm (Costa, Braga, Menezes, 2007). Among the category of fast algorithms, the methods use standard numerical optimization techniques such as conjugate gradient, quasi-Newton, and Levenberg-Marquard. The basic back propagation algorithm adjusts the weights in the steepest descent direction. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. Newtons method is an alternative to the conjugate gradient methods, which often converges faster. As a drawback, the method is complex and expensive for it's the Hessian matrix calculation in feed forward NNs. The computationally simpler quasi-Newton methods do not require calculation of second derivatives. Similarly, the Levenberg-Marquardt algorithm was also designed to approach second-order training speed without having to compute the Hessian matrix. There are a number of studies on different subjects that points the comparison between training algorithms (Ghaffari, Abdollahi, Khoshayand, Bozchalooi, Dadgar & Tehrani, 2006) (Srinivasulu & Jain, 2006) (Pereda, Lope & Maravall, 2006). Since Levenberg-Marquardt algorithm supplies faster convergence it is adopted and used in this article.

SOLAR RADIATION DATA FORECASTING RESULTS

In order to reduce computational complexity and to focus to the proposition, relatively short 1-D and 2-D prediction filters are used in this work. The filter templates are given in Fig. 2. These templates are also widely used in predictive image and signal coding.

For the minimum RMSE linear prediction, the optimal coefficients are analytically determined by solving Eq. 8. The 2-D image data is fed to the prediction system, and error figures are obtained for each hour. The error figure for 2-D 3-tap optimum filter is given in Fig. 3.

As a second step prediction model, two NN structures are applied to the data. In the first structure, the input is treated as 1-D, and the input network elements are i^{th}, $i+1^{st}$ and $i+2^{nd}$ elements of the data, where the output is the $i+3^{th}$ element for each sample in the data. In the second structure, the proposed 2-D image matrix form is used. The inputs of the networks are i,j^{th}, $i+1,j^{th}$ and $i,j+1^{st}$ elements of the 2-D data matrix and the output is $i+1, j+1^{st}$ element of the data matrix for each i and j. A 2-month period is used for testing.

The sigmoid function and the gradient descent algorithm with Levenberg-Marquard modification are used during learning process with three neurons at the hidden layer. To accelerate the speed of learning process a momentum term is used and is updated by a fraction of the previous weight update to the current one. After the learning phase, the network is simulated by the remaining image data and error samples are obtained (Fig. 4).

Root Mean Square Error (RMSE) values that are obtained from proposed optimum linear prediction filters and NNs are presented in Table I. The correlation coefficients between actual data values and predicted data values are also tabulated here.

FUTURE TRENDS

2-D representation has potential uses for different meteorological parameters and different models such as surface matching, clustering based classification, etc. Dynamical time varying behavior of the model may also be analyzed. Such analysis can be regarded as future works of this study.

Figure 2. 1-D and 2-D prediction templates used for modeling the image

To emphasize the efficiency of the proposed 2-D representation, two feed-forward NN structures, one for 1-D modeling and the other for the 2-D, are built and trained by the same data. The RMSE values are obtained as 42.012 and 38.66 for 1-D and 2-D case, respectively. This observation also justifies the efficiency of the 2-D data representation that exploits inter-day dependencies of the solar radiation pattern. Furthermore, it is clear that the 2-D NN structure provides better prediction than the optimum linear filter.

CONCLUSION

In this work, a novel approach is proposed for hourly solar radiation forecasting. The hourly solar radiation is interpreted and rendered as an 2-D image and its properties are examined. It is observed that two dimensional representations give more insight to the solar pattern than the regular 1-D interpretation. As an illustration, 1-D and 2-D optimal linear prediction filters with 3 coefficients are designed and compared in the sense of RMSE and correlation coefficients. The RMS energy value of the data and the prediction sequence are around 198. After applying the prediction, the RMS value of the prediction error reduces down to 44.33 using 1-D prediction. This value also constitutes the standard deviation of the statistical system. By using 2-D prediction, this value is reduced further to 41.09.

REFERENCES

Bulut H & Buyukalaca O. (2007) Simple model for the generation of daily global solar-radiation data in Turkey. *Applied Energy*, 84 (5), 477-491.

Costa M.A, Braga A.D & de Menezes B.R. (2007) Improving generalization of MLPs with sliding mode control and the Levenberg-Marquardt algorithm. *Neurocomputing*, 70 (7-9), 1342-1347.

Dorvlo, A.S.S. (2000) Fourier analysis of meteorological data for Seeb. *Energy Conversion and Management*, 41 (12), 1283-1291.

Elminir H.K, Azzam Y.A, Younes F.I. (2007) Prediction of hourly and daily diffuse fraction using neural network, as compared to linear regression models. *Energy*, 32 (8), 513-1523.

Figure 3. Error image obtained from 2-D optimal linear filter
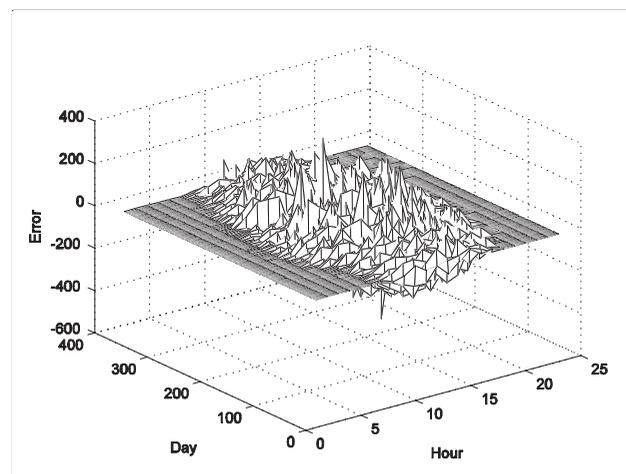
Figure 4. Test error image obtained from feed forward BP-NN
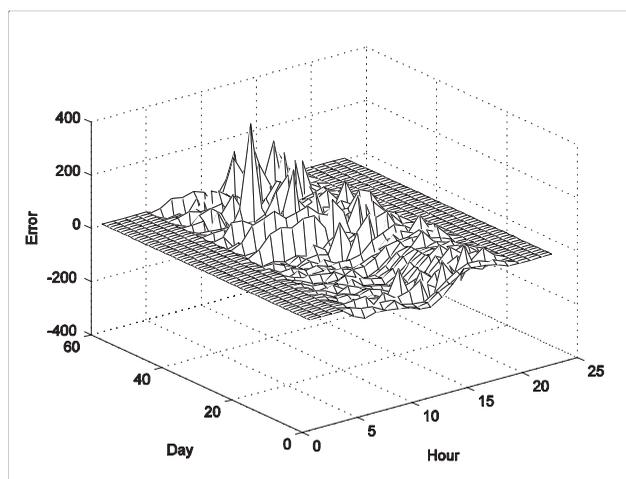
Table 1. RMSE values for proposed structures and Autocorrelation coefficients between actual values and predicted values of solar radiation data

	RMSE	RMSE for test data	R	R for test data
1-D Lin. Filter	44.33	-	0.963	-
2-D Lin. Filter	41.09	-	0.968	-
NN1 1-D	45.12	42.01	0.963	0.973
NN2 2-D	39.17	38.66	0.971	0.976

Hocaoglu F. O., Gerek O. N., & Kurban M. (2007) A novel 2-D model approach for the prediction of hourly solar radiation. *Lecture Notes in Computer Science*, 4507, 749-756.

Ghaffari A., Abdollahi H., Khoshayand M.R., Soltani Bozchalooi I., Dadgar A., & Rafiee-Tehrani M., (2006) Performance comparison of neural network training algorithms in modeling of bimodal drug delivery. *International Journal of Pharmaceutics*, 327 (1-2), 126-138.

Gonzalez R.C., Woods R.E., (2002) *Digital Image Processing*. Pearson Prentice Hall.

Jain P.K., & Lungu E.M.,(2002) Stochastic models for sunshine duration and solar irradiation. *Renewable Energy*, 27 (2), 197-209.

Kaplanis S.N.(2006) New methodologies to estimate the hourly global solar radiation; Comparisons with existing models. *Renewable Energy*, 31 (6), 781-790.

Sayood K.,(2000) *introduction to data compression* (2nd ed.). Academic Press.

Markvart T, Fragaki A., & Ross J.N.,(2006) PV system sizing using observed time series of solar radiation. *Solar Energy*, 80 (1), 46-50.

Mellit A., Benghanem M., Arab A.H., & Guessoum A., (2005) A simplified model for generating sequences of global solar radiation data for isolated sites: Using artificial neural network and a library of Markov transition matrices approach. *Solar Energy*, 79 (5), 469-482.

Muselli M., Poggi P., Notton G., & Louche A., (2001) First order Markov chain model for generating synthetic "typical days" series of global irradiation in order to design photovoltaic stand alone systems. *Energy Conversion and Management*, 42 (6), 675-687.

Pereda J, de Lope J., & Maravall D., (2006) Comparative analysis of neural network training methods for

inverse kinematics learning. *Lecture Notes in Artificial Intelligenc*, 4177, 171-179.

Srinivasulu S., Jain A., (2006) A comparative analysis of training methods for artificial neural network rainfall-runoff models. *Applied Soft Computing*, 6 (3), 295-306.

Ulgen K., & Hepbasli A. (2002) Prediction of solar radiation parameters through clearness index for Izmir, Turkey. *Energy Sources*, 24 (8), 773-785.

KEY TERMS

2-D Data Representation: A matrix containing vertical and horizontal indexes can also be considered as a 2-D image. A 2-D representation does not have to correspond to an image acquired by a camera or an imaging device. Here, the representation is used for the compact visualization of the solar data.

Artificial Neural Networks: A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Backpropagation Algorithm: Learning algorithm of ANNs, based on minimising the error obtained from the comparison between the outputs that the network gives after the application of a set of network inputs and the outputs it should give (the desired outputs).

Optimal Coefficient Linear Filters: A linear predictor takes a linear combination of past values in a time series, and assigns this combination as the prediction value. While taking the linear combination, the scales of each past sample should be calculated in a way that the prediction error has minimum amount of energy. Such a set of scales are called optimal coefficients of a linear filter.

Prediction Error: Difference between the actually measured and previously forcasted value of a time-series data. Commonly represented in terms of RMSE.

RMSE: Root-Mean-Squared Error. A quantitative error measure that defines the error between two sets of data as one-by-one differencing, squaring each difference, adding the squared terms, and finally taking the square root.

Solar Radiation: Radiant energy emitted by the sun from a nuclear fusion reaction that creates electromagnetic energy.

Speech–Based Clinical Diagnostic Systems

Jesús Bernardino Alonso Hernández
University of Las Palmas de Gran Canaria, Spain

Patricia Henríquez Rodríguez
University of Las Palmas de Gran Canaria, Spain

INTRODUCTION

It is possible to implement help systems for diagnosis oriented to the evaluation of the **fonator system** using speech signal, by means of techniques based on expert systems. The application of these techniques allows the early detection of alterations in the **fonator system** or the temporary evaluation of patients with certain treatment, to mention some examples. The procedure of measuring the **voice quality** of a speaker from a digital recording consists of quantifying different acoustic characteristics of speech, which makes it possible to compare it with certain reference patterns, identified previously by a "**clinical expert**".

A speech acoustic quality measurement based on an auditory assessment is very hard to assess as a comparative reference amongst different voices and different human experts carrying out the assessment or evaluation.

In the current bibliography, some attempts have been made to obtain objective measures of speech quality by means of multidimensional clinical measurements based on auditory methods. Well-known examples are: GRBAS scale from Japon (Hirano, M.,1981) and its extension developed and applied in Europe (Dejonckere, P. H. Remacle, M. Fresnel-Elbaz, E. Woisard, V. Crevier-Buchman, L. Millet, B.,1996), a set of perceptual and acoustic characteristics in Sweden (Hammarberg, B. & Gauffin, J., 1995), a set of phonetics characteristics with added information about the excitement of the vocal tract. The aim of these (quality speech measurements) procedures is to obtain an objective measurement from a subjective evaluation.

There exist different works in which objective measurements of speech quality obtained from a recording are proposed (Alonso J. B.,2006), (Boyanov, B & Hadjitodorov, S., 1997),(Hansen, J.H.L., Gavidia-Ceballos, L. & Kaiser, J.F., 1998),(Stefan Hadjitodorov & Petar Mitev, 2002),(Michaelis D.; Frohlich M. & Strube H.

W. ,1998),(Boyanov B., Doskov D., Mitev P., Hadjitodorov S. & Teston B.,2000),(Godino-Llorente, J.I.; Aguilera-Navarro, S. & Gomez-Vilda, P. , 2000).

In these works a voiced **sustained sound** (usually a vowel) is recorded and then used to compute speech quality measurements. The utilization of a voiced **sustained sound** is due to the fact that during the production of this kind of sound, the speech system uses almost all its mechanisms (glottal flow of constant air, vocal folds vibration in a continuous way, …), enabling us to detect any anomaly in these mechanisms. In these works different sets of measurements are suggested in order to quantify speech quality objectively. In all these works one important fact is revealed; it is necessary to obtain different measurements of the speech signal in order to compile the different aspects of acoustic characteristics of the speech signal.

BACKGROUND

A **speech recording** gives different characteristics of the speech quality of a speaker. The recorded speech signal can be represented in different domains. Each domain shows some of the speech characteristics in a preferential way. The main domains studied in speech processing are:

- *Time Domain*
- *Spectral Domain*
- *Cepstral Domain*
- *Inverse Model Domain*

Most works in digital speech signal processing are based on these domains. However, other works use new domains derived from the former ones.

In the following section the most important features of each domain are described.

Time Domain

A high quality speech signal possesses a more regular envolope than a low quality speech signal. This fact is more evident in short time intervals. The main phenomena that enable us to distinguish between high quality speech and low quality speech are:

The energy of the speech signal in a short time interval changes considerably between two consecutive intervals in low quality speech whereas in high quality speech there is a less change in energy.

In low quality speech unperiodicity (without perdiodicity) intervals during voiced sustained speech appear.

Spectral Domain

A low quality speech (a voiced **sustained sound**) has the following characteristics:

- Less regularity of the spectral envelope, mainly in low frequencies.
- More percentage of energy in low frequencies with regard to the total energy.
- Energy blocks in high frequencies. These blocks are caused by glottal noise.
- A great change of the power spectrum from a frame with regard to contiguous frames.

Figure 1. Speech signal in time domain: the five Spanish and sustained vowels are illustrated. The upper figure is a speaker with high quality speech. The lower figure is a speaker with low quality speech.
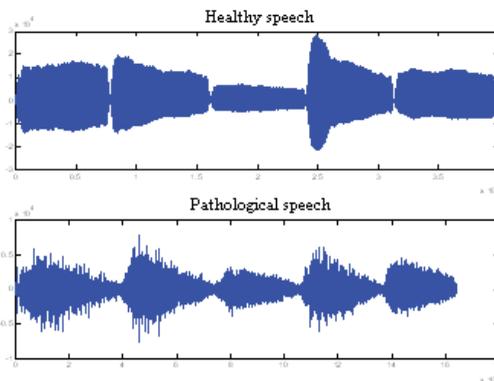
Figure 2. Sustained voiced sound during a short time interval from a high quality speech (left) and from a low quality speech (right).
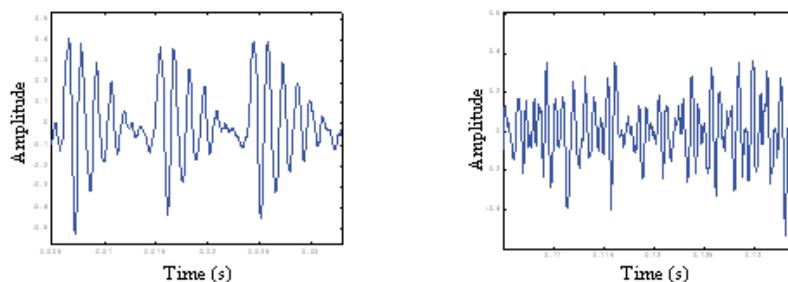
Figure 3. Estimated specgram from a high quality speech (top) and from a low quality speech (bottom). The five Spanish vowels are pronounced. The sample frequency is 22050 Hz.
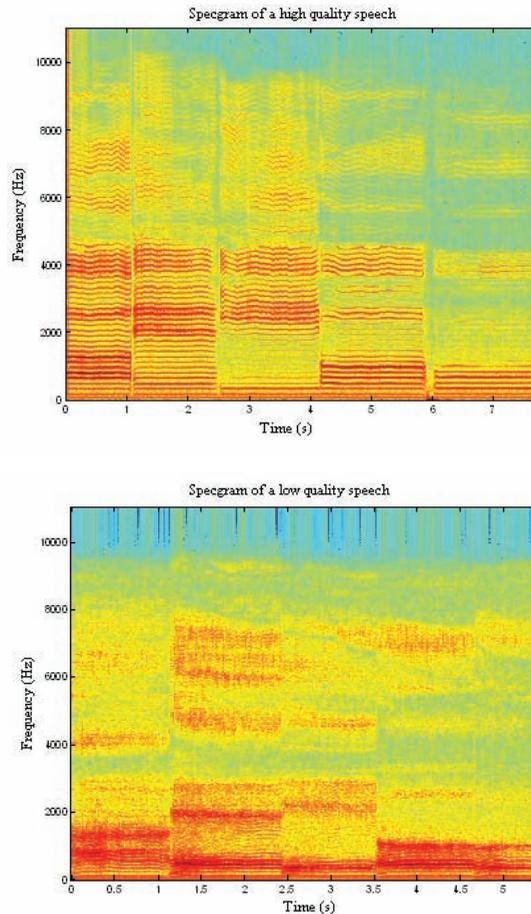
High quality speech concentrates its energy around certain formants, mainly the first and the third formants, whereas low quality speech has noise components around the formants.

High quality speech has great spectral wealth. However, low quality speech has a little amount of harmonic component, mainly concentrated in very low frequencies.

The amount of spectral wealth is a characteristic of the voice of a certain speaker. However, the spectral wealth variation in time (during the production of a sustained voiced sound) is indeed an indicator of low quality speech.

Another characteristic in low quality speech (during the production of a sustained voiced sound) is its vari-ations in vibration rhythm of vocal folds, i.e. frequency variation in *pitch* frequency.

Cepstral Domain

Characteristics to evaluate the speech quality can be identified in the cepstral domain: envelope of the spectrum, spectral wealth, harmonics and noise components identification, etc. A sustained voiced sound with three times the pitch period in length is used in the cepstral domain.

The spectral wealth of a speech signal can be quantified by the amplitude and width of the cepstral component of the *pitch* frequency. The existence of a peak with great amplitude indicates the presence of notorious energy in that harmonic component. This

Figure 4. A high quality speech (top) and a low quality speech (bottom) in the corrected power cepstrum.
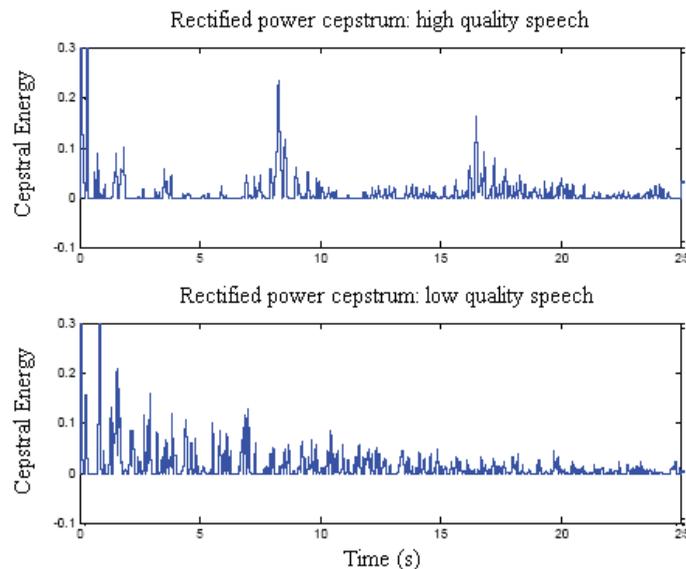
is a characteristic of high quality speech. A reduced width of the cepstral peak corresponding to the pitch indicates the high stability of the *pitch* frequency for three consecutive periods of pitch. This is also a characteristic of high quality speech. Characteristics such as amplitude and width of cepstral peak corresponding to the second harmonic can be also used to distinguish between high and low quality speech. High quality speech possesses a cepstral peak corresponding to the first harmonic narrower than the cepstral peak corresponding to the second harmonic.

Glottal noise in speech signal can be estimated by means of the relationships among different regions in the cepstral domain: the harmonic component (ceptrals components of *pitch* and its harmonics) and noise component (the remaining ceptrals components).

Inverse Model Domain

In this domain, the waveform of air pulse produced by the vocal folds during the production of a sustained voiced sound is estimated. The air pulse is also called residual signal or glottal flux. The estimatation is obtained with an inverse filter applied to the speech signal in order to eliminate the vocal tract effect and lips radiation effect.

The quality of speech can be quantified by some features of the glottal signal such as values of amplitude, time in which vocal folds start to open, time in which vocal folds are completely open, time in which the closing phase of vocal folds starts and different relationships between different times in the glottal cycle: *open quotient*, *speed quotient*, *closing quotient*, etc.

Non Linear Domain

The main comercial systems to evaluate speech quality from a recording objectively (Dr Speech *(Tiger Elemetric),* SSVA *(System for Sigle Voice Analysis),* MDVP *(Multi-Dimensional Voice Program)* ,EVA *(Evaluation Vocal Assistee),* CSL *(Computerized Speech Laboratory)* PRAAT, VISHACSRE *(Computerized Speech Research Environment),* MEDIVOZ, etc) do not assess nonlinear characteristics in speech signal.

The most popular model of characterization of the production voice system is a time-variant system, based on linear acoustics theories. It consists of a source/filter model. The existence of variations in spectral amplitude

S

of speech signal in the fundamental frequency leads us to cvonsider a nonlinear behaviour of speech signal.

A fundamental frequency (fs) and a subharmpnic (fs/2) have been identified in the speech signal (Xuejing Sun & Yi Xu, 1995). A subharmonic effect is the amplitude modulation or/and the frequency modulation. Other authors indicate that 31% of speakers with pathological speech have **subharmonics** in speech. However, the existence of **subharmonics** has also been identified in high quality speech (Haben, Kost & Papagiannis ,2003), . It is estimated that 10.5 % of speakers with healthy speech have **subharmonics**, not being a symptom of an anomalous speech.

There exist two theories to justify the presence of **subharmonics**:

Titza theory (TItze, IR., 1994): **subharmonics** are due to mechanical or geometric asymmetries between vocal folds.

Svec theory (Svec JG, Schutte HK, Miller DG, 1996): subharmonic frequency is due to the combinations of two vibrational modes (biphonation: the presence of two main frequencies) whose frequencies have a 3:2 relation.

Nevertheless, both theories are the same according to (Neubauer J., Eysholdt P., Eysholdt U., Herzel H., 2001), where the authors point out that biphonation is due to asymmetry between the left and right-hand vocal folds or to desynchronization in the back-forth vibration, (Haben C.M., Kost K. & Papagiannis G., 2003). Assymmetry and desynchronization are caused by differences in masses and viscoelastic properties between the vocal folds. This can be modelled by nonlinear phonation. In (Ayache S., Maurice Ouaknine, Dejonkere P., Prindere P. & Giovanni A., 2004) nonlinear models are suggested in order to explain the effect of mucus viscosity of vocal folds (mucus in vocal folds surface generates superficial tension and causes adhesion).

In the traditional model of the vocal tract, sound wave propagation is assumed to be plain wave propagation. However, sound pressure measurements and volume variation measurements are better fitted to a nonlinear model of dynamics fluid. This stems from turbulences (or even periodic turbulences) produced by cavities between the vocal folds and the false vocal folds. This

turbulence excites the vocal tract in the closing phase of vocal folds.

Fractal dimension has been studied by some authors. They conclude that high quality speech signal has a low dimensionality. It is stated (Orlikoff R.F., Baken R. J., 2003), that the amount of alinearities in the speech system is an indicator of anormal phonation and it has been suggested that phase space dimensionality, used for the attractor characterization, could be related to the amount of mass of vocal folds.

QUALITY SPEECH QUANTIFICATION

In the previous section a description of the different features of speech signal in different domains has been given. Theses features permit us to evaluate the speech quality. Each feature characterizes a physical phenomenum that is involved in voice production. A physical phenomenum can appear in different domains. In this work a set of physical phenomena to make a correct documentation of **voice quality** has been identified. The four physical phenomena identified are:

Voice stability: this is the ability of a speaker to create a constant intesity air flux in order to excite the vocal folds (during a sustained voiced sound). This physical phenomenum is quantified from measurements of speech stability.

Spectral wealth: this is the ability to generate a periodical movement in the vocal folds (during a sustained voiced sound) and produce a voiced excitation of the vocal tract with a great amount of spectral components. This physical phenomenum is quantified computing the pitch frequency stability and by the number of harmonics with high energy in different frequency bands.

Presence of noise: this is related to the presence of glottal noise in speech signal during the phonation of a sustained voiced sound. The presence of glottal noise is due to problems in the closing phase of vocal folds. This physical phenomenum is quantified by measuring the presence of nonstationary noise in speech signal.

Vocal folds irregularities: alinearities in speech system are caused by an anomalous working of vocal folds. This is due to irregularities in masses involved in closing phase of vocal folds, asymmetric movement

of vocal folds, and factors related to vocal folds mucus. These phenomena are quantified by means of nonlinear behaviour of speech signal.

An anormal speech shows unless one of the values corresponding to the quantification of the four physical phenomena is out of the normal range. This procedure of quality speech quantification permits us to identify anomalous speech qualities from diverse origins. These four kinds of physical phenomena can be quantified in different domains, existing different objective measurements of speech quality which are capable of quantifying with more or less accuracy a single physical phenomenum.

FUTURE TRENDS

In general, it is impossible to identify pathology in the **fonator system** using only a **speech recording**. This is stated by various authors. This stems from the fact that the acoustic characteristics of two speakers with different pathologies in the **fonator system** can be similar. Even in a visual inspection of the larynx the identity of the pathology cannot be determined. Furthermore, the coexistence of more pathology of the **fonator system** is also frequent.

Nevertheless, several works have focused on identifying the presence of anomalies in the **fonator system**. An automatic detection system of anomalies in speech system has the same diagram as a voice recognition system (see Figure 5).

The "Voice acquisition" block digitalizes speech signal. In this block, a discrimination between speech signal and noise is usually made and the segmentation of speech signal in frames is also carried out.

In the "Parameterization" block the speech quality is quantified using diverse quality measurements for each frame into which the speech signal is divided. The quantification enables us to identify differential characteristics among the different **classification units**.

In our case, the **classification units** are healthy and pathological speech. In this block, each speech frame is turned into a characteristics vector (or measurements vector). Some measurements average certain quantifications of an acoustic characteristic or evaluate its time evolution during the phonation.

An automatic classification of the characteristics vector is made in the "classification" block. The classification systems include Support Vector Machines, Neural Networks, etc.. In our case, the classification for each characteristics vector is between healthy speech and pathological speech.

We propose carrying out clinical studies in order to assess the usefulness of speech quality quantification automatic systems in speech therapy, otolaryngology and phoniatry. These studies will permit the application of the proposed protocol to measure the speech quality in fields such as the assessment of a surgical operation, documentation of a treatment evolution, medical-legal documentation and telemedicine.

It will be possible to implement automatic classification systems between healthy and pathological speech from databases with different qualities of speech, or even systems capable of automatically giving a measurement of the level of disphonia. These systems can be used in a screening evaluation or in a speech therapist evaluation.

CONCLUSION

In this work, the different physical phenomena which characterize **voice quality** have been identified. These phenomena have been quantified in order to obtain a correct documentation of **voice quality**. The quantification of nonlinear behaviour in signal speech has been introduced to describe in a more realistic way the vocal folds behaviour.

Voice quality quantification allows for the implementation of systems to help in pathologies diagnosis in

Figure 5. Diagram block of the automatic detection system of anomalies in the fonator system
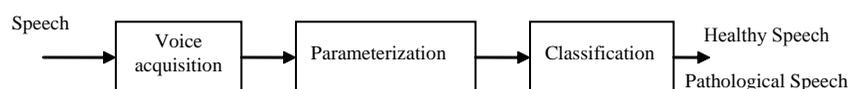
the **fonator system** by means of supervised automatic recognition systems such as Support Vector Machines (SVM) or Neural Networks (NN).

Advances in voice quantification applied to the voice synthesis field will improve naturality in the production of synthetic voices. The development of automatic mood detection is a possibility (for example, detection of sadness, anger or happiness) with the application of the knowledge acquired in measurements of voice quality. With these systems it will be possible to perceive no verbal language. These systems can be applied to new generations of human-computer interfaces.

REFERENCES

Ayache S.,Ouaknine M., Dejonkere P., Prindere P. & Giovanni A. (2004), "Experimental Study of the Effects od Surface Mucus Viscosity on the Glottic Cycle", Journal of Voice, 18(1): 334-340.

Boyanov B., Doskov D., Mitev P., Hadjitodorov S. & Teston B. (2000) , New cepstral parameters for description of pathologic voice, *Comptes Rendus de L'Academie Bulgare des Sciences* (Ann. of Bulgarian Academy of Sciences), 53(3), 41-44.

Boyanov, B & Hadjitodorov S. (1997), Acoustic analysis of pathological voices. A voice analysis system for the screening of laryngeal diseases, *IEEE Engineering in Medicine and Biology Magazine*, 16(4), 74 – 82.

Dejonckere, P. H. Remacle, M. Fresnel-Elbaz, E. Woisard, V. Crevier-Buchman, L. Millet, B. (1996), "Differentiated perceptual evaluation of pathological voice quality: reliability and correlations with acoustic measurements". *Revue de Laryngologie Otologie Rhinologie*, 117(2), 219-224.

Godino-Llorente, J.I.; Aguilera-Navarro, S. & Gomez-Vilda, P. (2000), Non supervised neural net applied to the detection of voice impairment. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP '00. vol. 6, pp.3594-3597.

Haben C. M., Kost K. & Papagiannis G. (2003), "Lateral Phase Mucosal Asymmetries in the Clinical Voice Laboratory", Journal of Voice, 17(1), 3-11.

Hadjitodorov S. & Mitev P. (2002), A computer system for acoustic analysis of pathological voices and laryngeal diseases screening, *Medical Engineering Physics*, (24):419-429.

Hammarberg, B. & Gauffin, J. (1995), *Perceptual and acoustic characteristics of quality differences in pathological voices as related to physiological aspects*,. in O. Fujimura & M.Hirano (eds.), Vocal Fold Physiology,283-303.

Hansen, J.H.L.; Gavidia-Ceballos, L. & Kaiser, J.F. (1998), A nonlinear operator-based speech feature analysis method with application to vocal fold pathology assessment, *IEEE Transactions on Biomedical Engineering*, 45(3), 300-313.

Hirano, M. (1981), *Clinical Examination of Voice*, New York, Springer-Verlag.Dejonckere, P. H. Remacle, M. Fresnel-Elbaz, E. Woisard, V. Crevier-Buchman, L.

Laver, J. (1991), *The Gift of Speech*, Edinburgh University Press

Michaelis D.; Frohlich M. & Strube H. W. (1998), Selection and combination of acoustic features for the description of pathologic voices. *Acoustical Society of America*. 103(3),1628-1640.

Millet, B. (1996), Differentiated perceptual evaluation of pathological voice quality: reliability and correlations with acoustic measurements. *Revue de Laryngologie Otologie Rhinologie*, 117(2), 219-224.

Neubauer J., Eysholdt P., Eysholdt U. & Herzel H.(2001), "Spatio-temporal analysis of irregular vocal fold oscillations: Biphonation due to desynchronization of spatial models", J. Acoustical Society of America, 110 (6), 3179-3192.

Orlikoff R.F., Baken R. J., (2003), Curing Diagnosis: Improving the Taxonomy of Phonatory Dysfunction, *Sixth Conference on Advances in Quantitative Laryngology. Hamburg*, Germany.

Robert F. Orlikoff & R. J. Baken (2003), *Curing Diagnosis: Improving the Taxonomy of Phonatory Dysfunction*, Sixth Conference on Advances in Quantitative Laryngology. Hamburg, Germany.

Svec JG, Schutte HK & Miller DG (1996)." A Subharmonic vibratory pattern in normal vocal folds", Journal of Speech and Hearing Research, 39(1), pp.135-143.

Titze, IR. (1994), Principles of Voice Production, Englewood Cliffs, NJ: Prentice-Hall, Inc.

Xuejing Sun & Yi Xu (1995), Perceived Pitch of Synthesized Voice with Alternate Cycles, *Journal of Voice*, 16(4) 443-459.

KEY TERMS

Characterization or Representation Domains: These are the different spaces into which a signal can be transformed, where certain characteristics of this signal (levels of regularity, levels of noise, similarities, etc) are pronounced of preferential form.

Diagnosis Automatic Intelligent Systems: These are systems which enable the identification of pathological states without the presence of a **clinical expert**. These systems are oriented to preventive medicine or first screening.

Disphonia: This is the alteration of voice quality. It is mainly caused by laryngeal pathologies. Other different motives to those of a medical nature can produce changes in voice quality, such as, for example, factors related to mood.

GRBAS: Objective measures of speech quality by means of multidimensional clinical measurements based on auditory methods.

Help Systems for Diagnosis: These are systems that help the clinical professionals to identify certain situations that need special attention. They are used generally in tasks of clinical monitorization.

Laryngeal Pathology: Due to different organic injuries (such as malformations, benign injury, inflammations, infections, precancerous and cancerous injuries, traumatisms, or endocrine, neurological and auditive injuries), different functional disphonies (in spoken and sung voice) and of psychiatric origin.

Pitch: Vibration frequency of vocal folds. In fact, there is not complete periodicity in the vibration of vocal folds. That is why it is said that vocal folds have a quasiperiodic movement.

State of the Art in Writer's Off-Line Identification

Carlos M. Travieso González
University of Las Palmas de Gran Canaria, Spain

Carlos F. Romero
University of Las Palmas de Gran Canaria, Spain

INTRODUCTION

Today, advances in Computer Science and the proliferation of computers in modern society are an unquestionable fact. Nevertheless, the continuing importance of orthography and the hand-written document are also beyond doubt.

The new technologies permit us to work with on-line information collecting, but there is still a large quantity of information in our society which requires using algorithms for samples off-line. Security in certain applications requires having biometric systems for their identification; in particular, banking checks, wills, postcards, invoices, medical prescriptions, etc, require the identity of the person who has written them to be verified. The only way to do this is with writer recognition techniques.

Furthermore, many hand-written documents are vulnerable to possible forgeries, deformations or copies, and generally, to illicit misuse. Therefore, a high percentage of routine work is carried out by experts and professionals in this field, whose task is to certify and to judge the authenticity or falsehood of handwritten documents (for example: wills) in a judicial procedure. Therefore nowadays research on writer identification is an active field.

At present, some software tools enable certain characteristics to be displayed and visualised by experts and professionals, but these experts need to devote a great deal of time to such investigations before they are able to draw up conclusions about a given body of writing. Therefore, these tools are not time-saving and nor do they provide a meticulous analysis of the writing. They have to work with graph paper and templates in order to obtain parameters (angles, dimensions of the line, directions, parallelisms, curvatures, alignments, etc.). Moreover, they have to use a magnifying glass and graph paper in order to measure angles and lines. This research aims to lighten this arduous task.

BACKGROUND

Writer identification is possible because the writing for each person is different, and everyone has intrinsic characteristics. The scientific bases for this idea come from the human brain. If we attempt to write with our less skilful hand, there will be some parts or strokes very similar to the writing which we make using our skilful hand. This is because the brain sends the commands for carrying out the writing and not the hands.

Generally, this effect is projected toward the writing by two types of forces, which are:

- Conscious or known: because it is controlled by the individual's own free will.
- Unconscious: because it escapes the control of the individual's own free will. This is divided into forces of mechanical and emotional means, which behaviour feelings.

Everybody writes using their brain, and simultaneously the handwritten impulse, which is the symbolism of the space in order to obtain the dimensions of the writing, is adapted proportionally, the size of the text being maintained or modified depending on whether the individual is forced to write in a reduced space.

Nowadays, writer identification is a great challenge because such research work has not been as fully developed as that of identification based on fingerprints, hands, face or iris (other biometric techniques), due mainly to the fact that the operation of the brain is very difficult of parameterize. On the other hand, the

above-mentioned techniques use widely researched biometric information.

Most of the characteristics implemented offer information in the vertical and horizontal plane (Zhenyu, Bin, Jianwei, Yuan, & Xinge, 2005) (Zhenyu, Yuan, & Xinge, 2005) (Schlapbach, & Bunke, 2006) (Bulacu, & Schomaker, 2005). We have introduced a new parameter, the proportionality index, which projects in all directions, depending on the selected points.

OFF-LINE WRITER IDENTIFICATION SYSTEM

As with the majority of the works proposed to the present date on biometric recognition, the framework of the system depends on the basic steps showed in figure 1. The images acquisition is a previous step to this system; therefore, this system is an off-line system. The data have to be scanned or photographed in order to build our database.

Data Acquisition

The forensic analysis of hand-written documents requires an extensive database of a known writer's hand-written samples. Therefore samples are gathered of different writers' writing and in turn several samples are taken of each one owing to the temporary invariability.

The creation conditions of a database have to be normalized with different types of paper, pen, and similar place of support (for doing the writing) because our work is centred on the writing and the efficiency of proposed parameters. For these off-line systems, the documents have been generated, and therefore, for the building of the database, the system has to be scanned or a high resolution picture taken. 300dpi on grey scale (8 bits) is a good threshold.

Image Pre-Processing and Segmentation

The first step of the image pre-processing consists of utilizing Otsu's method (or another method), which permits us to determine the necessary grey threshold value to carry out the binarization of the samples (Otsu, 1979).

As result of the binarization, in most cases, the line of writing remains with irregular appearance. For this reason, another pre-processing step is carried out, which enables the line to be smoothed out, thus remaining well defined. This also eliminates the existing noise in the images after the scanning process.

As previous step to the separation of words or connected components, the detection and elimination of the punctuation marks (full stops, accents and commas etc.) is carried out.

Finally, the words which compose the lines of writing are segmented (baselines) and for this, it is necessary to establish limits for each of the words. For this

Figure 1. System of writer identification
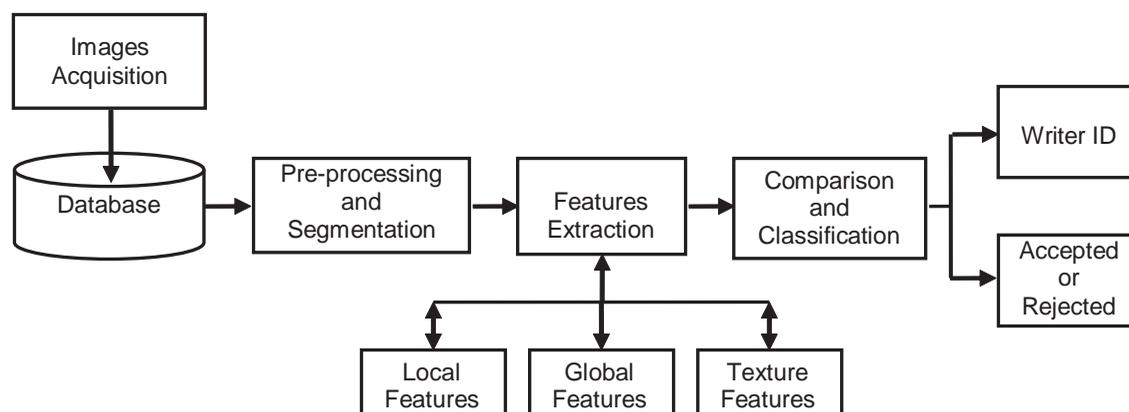
estimation, the method of the "Enclosed Boxes" (Ha, Haralick & Phillips, 1995) was used, which provides us the coordinates that will allow us to segment the words. The enclosed boxes are defined as the minimum rectangle that contains the connected component.

Feature Extraction

The calligraphic expert's task is usually to make a statistical list of different quantitative and qualitative measurements carried out on the document in question, and to present this as evidence in a judgment. These include order, legibility, construction of letters, connection, dimension, slant of the letters, space among words and among characters, alignment and skew of the baseline, initial and final stroke, continuity of the stroke, punctuation, control and movement of the ball-point pen.

The developed systems compare the document being tested with the samples of the database, using image digital processing in order to extract the features defined by the system.

We can define three different kinds of features, local, global and texture features (see figure 1).

The local features examine the construction of the individual characters to identify details of certain letters, since it is considered that it is very difficult for a writer to change the way of writing her/his letters. One of the techniques consists of dividing into regions the images of the segmented letters and then for each region to calculate the direction of the gradient; also a geometric description can be obtained by analysing the presence of corners, diagonal, vertical and horizontal lines, direction and angle of the edges and hinges (Zhang, Srihari, & Lee, 2003).

Another way to describe letters is through the pair of coordinated (x,y) of the contour of the connected components and as each writer is considered a generator of a finite number of basic patterns formed by these connected components; it can be characterized by the discrete probability density function of emission of a basic pattern of the strokes (Schomaker, & Bulacu, 2004). Another similar method detects the morphological invariants using an automatic classifier of grapheme; in (Bensefia, Pasquet, & Heutte, 2002), the authors have shown that the variability of the writing can be measured through these invariants because each writer writes the same letters using such patterns or graphemes.

The global features try to describe the properties of the writing and they are statistical measurements extracted from the whole sample of the handwritten document, paragraphs, lines and words to identify (Grening, Sagar, & Leedham, 2005) (Tomai, Zhang, & Srihari, 2004) (Marti, Messerli, & Bunke, 2001). In (Wirotius, Seropia, & Vicent, 2003) a study was carried out on the distribution of gray levels in the pixels of the stroke, calculating the curve of evolution of these levels along sections of the stroke observing that the symmetry with respect to the minimum of the curve presents a great variability according to the writer and the way in which the ball-point pen is located on the paper.

In (Srihari, Cha, Arora, & Lee, 2002) the variation of gray levels is detected by means of its entropy, giving an idea of the pressure applied when writing. Another measurement that provides information of pressure, thickness of the stroke and size of the writing is to count the number of black pixels of the binarized image, which can also allow the movement of the ball-point pen when writing to be estimated indirectly, by means of the quantity average of internal and external contours.

As the contours consist of connected pixel segments, they can be stored as a Chaincode representation where their vertical, horizontal and diagonals components will represent the formation of the stroke.

Other global features are the average slant (Bonzinovic, & Srihari, 1989), localization of the baselines and their skew, height of the ascending, descending and middle body of writing (Marti, Messerli, & Bunke, 2001) (Romero, Travieso, Alonso, & Ferrer, 2007), average width of the characters, behavior of the margins, length of the words and distance between lines and words.

In order to obtain the texture features, the writing sample is viewed as a simple image and not as a manuscript, and therefore each person's writing can be considered as a different texture; applying to it filters of Gabor and co-occurrence Matrixes (Said, Peake, Tan & Baker 1998) for example.

In order for features to represent the writing style, they must fulfill the following requirement: the fluctuations in an individual's writing must be as small as possible, while the fluctuations among different writers must be as great as possible. Each one of these features is evaluated to determine their discrimination index

which allows the utility of each feature to be measured for the identification of writers.

One of the biggest difficulties for automatic identification is the handling of a great variability of writing styles, and there is therefore still some work to be done in the feature extraction stage, since the purpose of this stage is to detect the discriminate features of the writing that characterize the styles of people's writing. Up to now the majority of the characteristics used by the experts are not as yet algorithmically implemented.

In this present work, a list has been created of geometrical parameters of different measurements to analyse documents. In order for the characteristics to represent the style of writing, they should comply with the following requirement: the fluctuations in the writing of a person should be as small as possible, while the fluctuations among different writers should be as large as possible.

This characteristic is included in the list of the following characteristics already developed (Romero, Travieso, Alonso, & Ferrer, 2006) (Hertel, & Bunke, 2003):

- length of the words,
- quantity of pixels in black,
- estimation of the width of the letters,
- height of the medium body of writing,
- heights of the ascending and descending,
- height relation between of the ascending and medium body,
- height relation between descending and medium body,
- height relation between descending and ascending,
- height relation between medium body and the wide of writing.

The quantity of black pixels and the long words will give us an estimation of the dimension and thickness of the line, the width of letters and the height of the medium body. Besides these are the distinctive characteristics of the style of writing.

The estimation of the width of letters is carried out by seeking the row with the greatest quantity of transition of black to white (0 to 1). The number of white pixels between each transition is counted and this result is averaged.

To measure the height of the medium body of the words, the goal is to determine the upper and lower baseline through maximum and minimum values, and to measure the distance between them.

To approach the baselines of each word, it was decided to use the adjustment of minimum mean square error that is based on finding the equation (see expression 1) that is best adjusted to a set of points "n" (Chin, Harvey, & Jennings, 1997). The equation is the following:

$$y = ax + b \tag{1}$$

where the coefficients "*a*" and "*b*" determine the lineal polynomial regression by means of the following expressions:

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \tag{2}$$

$$b = \frac{\sum_{i=1}^{n} y_i - a\sum_{i=1}^{n} x_i}{n} \tag{3}$$

Figure 2. Zones and baselines
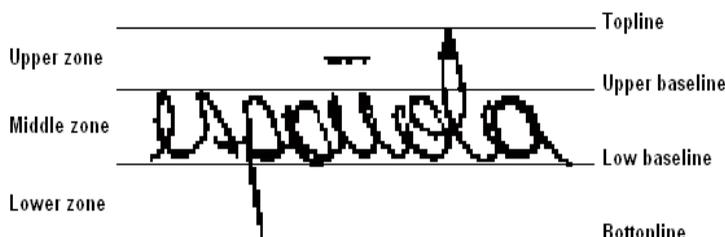
Those values of "*a*" and "*b*", based on the coordinates of minimums or maximums detected in the contour of the word, are different baselines. Minimums are to approach the lower baseline and the maximums for the upper baseline.

The extraction of the proportionality index is a new parameter in our system and in the references. The selection of the points is of random form but with some indications, and therefore we have located the most representative sites as being ascending ones, descendent, terminations, etc. In this paper the most representative points (red points) are displayed in figure 3. For this same word for each writer, we have marked the same red points.

The marked points are united (see Figure 3), and each line between two points is considered as a segment. We have measured the Euclidean length of each segment obtaining a mean and a standard deviation. These are new and novel parameters, which provide information from every direction of a word.

Classification System

The problem of the identification of writers can be seen according to two different approaches (see Figure 4); the first approach is the verification that allows us to determine whether two documents were written by the same person or by two different people.

The second approach is the identification that consists of recognising a writer among a set of N candidates. This case can be seen as a problem of classification of N classes. Due to the potentially great number of candidates, the decision is based on the measurement of the nearest neighbour; its advantage is that it identifies the writer directly.

Both approaches resort to some method of similarity measurement or distances between the samples; and the system must be trained with a set of handwritten samples belonging to each candidate (supervised classification). The most commonly used classification methods are nearest k-neighbours (Hertel, & Bunke, 2003), Neuro-

Figure 3. Segments obtained when points are united (proportionality index)
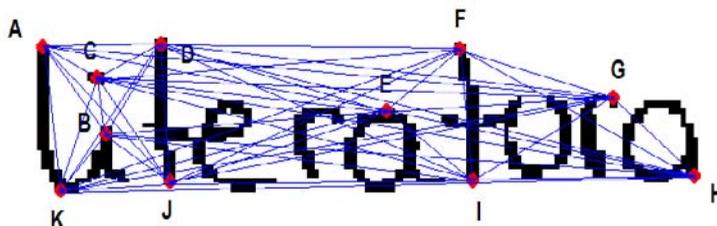
Figure 4. Difference between identification and verification
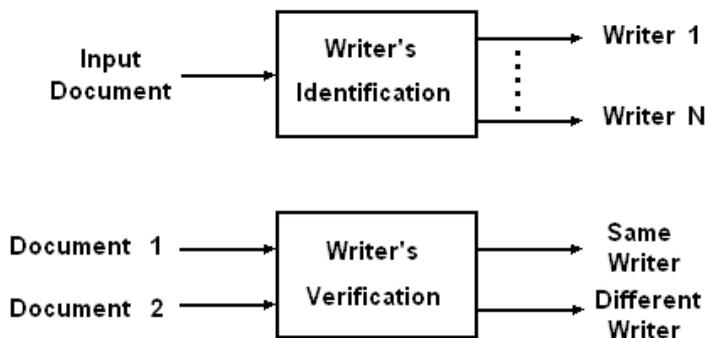
nal Network (Bishop, 1995), Hidden Models Markov (Juang, & Rabiner, 1992), Gaussian Mixture Models (Schlapbach, & Bunke, 2006), etc.

In the following table, we can see a comparison of different methods, showing the type of samples, number of writers and its success rates.

FUTURE TRENDS

The future tendency is the use of the Graphology, as fundamental element to establish new features, because experts or professionals have more years of experience and credibility in this field. The quantitative implementation of these features will result in positive growth in terms of success rates.

CONCLUSION

Writer identification systems could end up as a powerful tool to help Calligraphic Experts, since it will allow them to reduce the time invested in the analysis of certain features of the writing, since the system will develop the detection, extraction, comparison, classification and recognition tasks. Logically, it will not replace them because it is impossible to equal the observation power of the human in this complex application; therefore there will always be features that will have to be analyzed by the experts because they are very difficult to implement algorithmically.

Finally, off-line writer identification is an open research field, where the operation of different and new methods is both improving and spreading in terms of usage.

REFERENCES

Zhenyu, H., Bin, F., Jianwei, D., Yuan, Y.T., & Xinge, Y., (2005), A novel method for offline handwriting-based writer identification, Proceedings of Eighth International Conference on Document Analysis and Recognition, 1, 242 – 246.

Zhenyu, H., Yuan, Y.T., & Xinge, Y., (2005), A contourlet-based method for writer identification, IEEE International Conference on Systems, Man and Cybernetics, 1, 364 – 368.

Schlapbach, A., & Bunke, H., (2006), Off-lineWriter Identification Using Gaussian Mixture Models, 18th International Conference on Pattern Recognition, 3, 992 – 995.

Bulacu, M., & Schomaker, L., (2005), A comparison of clustering methods for writer identification and verification, Eighth International Conference on Document Analysis and Recognition, 2, 1275 – 1279.

Otsu, N., (1979), A threshold selection method from gray-level histograms, IEEE Transaction on Systems, Man and Cybernetics, 9(1), 62-66.

Table 1. Comparison of different published methods

Author	Samples	# Writers	Success Rate
Marti et al., 2001	Handwritten Text	20	90,7 %
Srihari et al., 2002	Handwritten Text	100	82 %
		900	59 %
Schomaker et al., 2004	Paragraphs in Uppercase	150	95 %
Said et al., 1998	Handwritten Text	40	95 %
Hertel et al., 2003	Text Lines	50	90,7 %
Bunke et al., 2004	Text Lines	100	96,56 %
Zois et al., 2000	Words	50	92,5 %
Bensefia et al., 2005	Paragraphs	150	86 %
Romero et al., 2007	Words	50	97,00%

Ha, J., Haralick R.M., & Phillips, I.T., (1995), Document page decomposition by the bounding-box project, Proc. IEEE International Conference on Document Analysis and Recognition, 2, 1119.

Zhang, B., Srihari, S.N., & Lee, S., (2003), Individuality of Handwritten Characters, Proceeding of the 7th International Conference on Document Analysis and Recognition, 1086-1090.

Schomaker, L., & Bulacu, M., (2004), Automatic Writer Identification Using Connected-Component Contours and Edge-Based Features of Uppercase Western Script, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6), 787 – 798.

Bensefia, A., Pasquet, T., & Heutte, L., (2002), Writer Identification By Writer's Invariants, Proceeding of the 8th International Workshop on Frontier in Handwriting Recognition, 274-279.

Grening, C.M., Sagar, V.K., & Leedham, C.G., (2005), Handwriting Identification Using Global and Local Features for Forensic Purposes, European Convention on Security and Detection, 272-278.

Tomai, C., Zhang, B., & Srihari, S., (2004), Discriminatory Power of Handwritten Words for Writer Recognition, Proceedings of the 17th International Conference on Pattern Recognition, 638-641.

Marti, U.V., Messerli, R., & Bunke, H., (2001), Writer Identification Using Text Line Based Features, Sixth International Conference on Document Analysis and Recognition, 101-105.

Wirotius, M., Seropia, A., & Vicent, N., (2003), Writer Identification from Gray Level Distribution, Proceeding of the 7th International Conference on Document Analysis and Recognition, 1168-1172.

Srihari, S., Cha, S.H., Arora, H., & Lee, S., (2002), Individuality of Handwriting, Journal of Forensic Sciences, 47(4), 1-17.

Bonzinovic, R., & Srihari, S., (1989), Off-line Cursive Script Word Recognition, IEEE Transaction on Pattern Analysis and Machine Intelligence, 11(1), 68-73.

Romero, C.F., Travieso, C.M., Alonso, J.A., & Ferrer, M.A., (2007), Using Off-line Handwritten Text for Writer Identification, WSEAS Transactions on Signal Processing, 3(1), 2007.

Said, H.E., Peake, G.S., Tan T.N., & Baker K.D., (1998), Writer Identification from Non-uniformly Skewed Handwriting Images, Proceedings of the 9th British Machine Vision Conference, 478-487.

Hertel, C., & Bunke, H., (2003), A Set of Novel Features for Writer Identification, Workshop on Audio and Video Based Biometric Person Authentication, 679-687.

Romero, C.F., Travieso, C.M., Alonso, J.A., & Ferrer, M.A., (2006), Writer Identification by Handwritten Text Analysis, Proc. of the 5th WSEAS int. Conf. on System Science and Simulation in Engineering, 204-208.

Chin, W., Harvey, M., & Jennings, A., (1997), Skew Detection in Handwritten Scripts, IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications, 1, 319-322.

Bishop, C.M., (1995) Neural Networks for Pattern Recognition (Oxford University Press).

Juang, B.H., & Rabiner, L.R., (1992), Spectral representations for speech recognition by neural networks-a tutorial, Proceedings of the Workshop Neural Networks for Signal Processing, 214 – 222.

Marti, U.V., Messerli, R., & Bunke, H., (2001), Writer Identification Using Text Line Based Features, Sixth International Conference on Document Analysis and Recognition, 101-105.

Zois, E.N., & Anastassopoulus, V., (2000), Morphological Waveform Coding for Writer Identification, Pattern Recognition, 33(3), 385-398.

Bensefia, A., Pasquet, T., & Heutte, L., (2005) Handwritten Document Analysis for Automatic Writer Recognition. Electronic Letters on Computer Vision and Image Analysis, 72-86.

KEY TERMS

Biometric System: This is a system which identifies individuals using behaviour or physical characteristics.

Classification System: Learning algorithm which generates automatic results from a features input. This system generally has as many outputs as classes for classifying.

Feature Extraction: This is a process which is used to obtain certain characteristics which are intrinsic and discriminate of a thing.

Image Pre-Processing: Set of tools applied to the images in order to provide other improved images for other tasks.

Off-Line System: A system whose operation is based on data that have been acquired before of its operation.

On-Line System: A system whose operation is based on data which are acquired during its operation.

Supervised Classification: This is a system that generates a model using training samples with labels, and it uses that model in order to establish an evaluation or test with other samples without labels.

Writer Identification: The application of biometric identification by handwriting. Full texts or just several words can be used.

State-of-the-Art on Video-Based Face Recognition

Yan Yan
Tsinghua University, Beijing, China

Yu-Jin Zhang
Tsinghua University, Beijing, China

INTRODUCTION

Over the past few years, face recognition has gained many interests. Face recognition has become a popular area of research in computer vision and pattern recognition. The problem attracts researchers from different disciplines such as image processing, pattern recognition, neural networks, computer vision, and computer graphics (Zhao, Chellappa, Rosenfeld & Phillips, 2003).

Face recognition is a typical computer vision problem. The goal of computer vision is to understand the images of scenes, locate and identify objects, determine their structures, spatial arrangements and relationship with other objects (Shah, 2002). The main task of face recognition is to locate and identify the identity of people in the scene. Face recognition is also a challenging **pattern recognition** problem. The number of training samples of each face class is usually so small that it is hard to learn the distribution of each class. In addition, the within-class difference may be sometimes larger than the between-class difference due to variations in illumination, pose, expression, age, etc.

The availability of the feasible technologies brings face recognition many potential applications, such as in face ID, access control, security, surveillance, smart cards, law enforcement, face databases, multimedia management, human computer interaction, etc (Li & Jain, 2005).

Traditional still image-based face recognition has achieved great success in constrained environments. However, once the conditions (including illumination, pose, expression, age) change too much, the performance declines dramatically. The recent FRVT2002 (Face Recognition Vendor Test 2002) (Phillips, Grother, Micheals, Blackburn, Tabassi & Bone 2003) shows that the recognition performance of face images captured in an outdoor environment and different days is still not satisfying. Current still image-based face recognition algorithms are even far away from the capability of human perception system (Zhao, Chellappa, Rosenfeld & Phillips, 2003). On the other hand, psychology and physiology studies have shown that motion can help people for better face recognition (Knight & Johnston, 1997; O'Toole, Roark & Abdi, 2002). Torres (2004) pointed out that traditional still image-based face recognition confronts great challenges and difficulties. There are two potential ways to solve it: video-based face recognition technology and multi-modal identification technology. During the past several years, many research efforts have been concentrated on video-based face recognition. Compared with still image-based face recognition, true **video-based face recognition** algorithms that use both spatial and temporal information started only a few years ago (Zhao, Chellappa, Rosenfeld & Phillips, 2003).

This article gives an overview of most existing methods in the field of video-based face recognition and analyses their respective pros and cons. First, a general statement of face recognition is given. Then, most existing methods for video-based face recognition are briefly reviewed. Some future trends and conclusions are given in the end.

BACKGROUND

From a general point of view, a complete video-based face recognition system includes face detection module, face tracking module, feature extraction module and face recognition module. Face detection is at the bottom layer. The task of face detection is to determine the spatial position and pose of the face(s). Face tracking is at the middle layer. It follows the continuous change

Figure 1. A general framework of video-based face recognition system
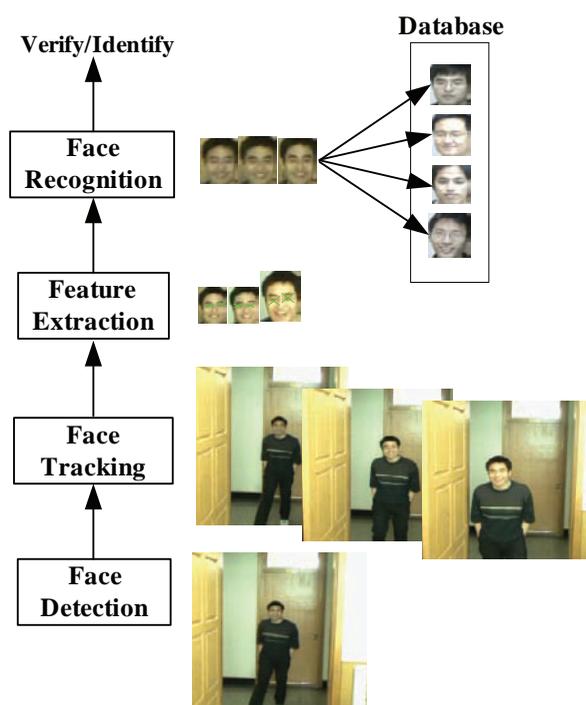
of face position over time. Feature extraction is at a higher layer. Its task is to locate the position of facial features such as eye, nose, etc, and pull out related information. Face recognition module is at the top layer. The face recognition module identifies or verifies the input face(s), with the help of databases. Figure 1 gives the general framework of video-based face recognition system, with a flowchart and some examples.

In this article, the focus will be on the top layer of face recognition systems—face recognition module. The general statement of **face recognition** can be defined as: given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces (Zhao, Chellappa, Rosenfeld & Phillips, 2003). The still image-based face recognition usually refers to the process in which the input is a still image. On the other side, the video-based face recognition often refers to the process in which the input is a shot of video. The database can be also still image(s) or video. Therefore, according to different modalities of the input and database, four different scenarios of face recognition can be distinguished. Table 1 shows these four different scenarios of face recognition. **Video-**

based face recognition usually refers to both "Video - Image(s)" face recognition and "Video - Video" face recognition, that is, with video input.

Compared with still image-based face recognition, video-based face recognition can utilize the temporal and spatial information available in the video. It's widely believed that video-based face recognition is more promising than still image-based face recognition. However, there also exist some difficulties in video-based face recognition, such as low-resolution face images, large variations of scale, illumination change, pose change, and occasionally occlusion in video. It is worth noting that if the time information of video is not considered, the video-based face recognition becomes the multiple-still-images input face recognition.

VIDEO-BASED FACE RECOGNITION

According to the classification shown in Table 1, four scenarios of face recognition will be reviewed separately. The emphases will be put on "Video - Image(s)" face recognition and "Video - Video" face recognition. For simplicity, the position of the face in the video is assumed to be known in advance.

"Image - Image(s)" Face Recognition

"Image - Image(s)" face recognition is the traditional still image-based face recognition. Numerous still image-based face recognition methods have been developed during the past few decades (Zhao, Chellappa, Rosenfeld & Phillips, 2003). Among them, global feature matching methods, such as Eigenface (Turk & Pentland, 1991), Fisherface (Belhumeour, Hespanha & Kriegman, 1997) and Bayesian (Moghaddam, Jebara & Pentland, 2000); and local feature matching methods, such as Elastic Bunch Graph Matching (EBGM) (Wiskott, Fellous, Krueuger & Malsburg, 1997), are the widely used face recognition approaches. Recently, 3D deformable models (Blanz & Vetter, 2003) and Local Binary Pattern (LBP) (Ahonen, Hadid & Pietikäinen, 2006) are the newly-emerging methods. Traditional still image-based face recognition has been widely used in biometric authentication, information security, etc.

Table 1. Four different scenarios of face recognition

Input \ Database	Still-Image(s)	Video
Still-Image	Image - Image(s)	Image - Video
Video	Video - Image(s)	Video - Video

"Image - Video" Face Recognition

"Image - Video" face recognition is to identify or verify a given face in the stored video sequences. "Image - Video" face recognition is also called human face-based video retrieval. Typical scenes includes finding suspects in the recorded surveillance video or finding a person in the film or news video from a given face image. Theoretically, it should do video preprocessing first, such as shot extraction. Then, face detection and tracking are performed to obtain the video shot of every face. Face recognition is conducted in the last step. Due to the complex scenes in such videos (film, news, surveillance video), most literature focuses on video preprocessing phase (Arandjelović & Zisserman, 2005b). In recent years, some scholars applied 3D model for television people retrieval (Everingham & Zisserman, 2004).

"Video - Image(s)" Face Recognition

"Video - Image(s)" face recognition can be formulated as follows: given a shot of video, identify or verify the face inside by using a still-image(s) database. With the wide-spread usage of video acquisition hardware, there exist many video sequences in the application of security authentication, video surveillance, etc. At the same time, most existing databases are still-image(s) database. Therefore, how to make better use of the input video is of important value in real applications.

Traditional approaches can be roughly divided into two categories: one is to perform face tracking until a facial image satisfies certain rules (such as size, pose). Then traditional still image-based face recognition methods are applied. The disadvantages of such approaches are the difficulty of defining the rules and not making full use of all information in the video. Another is to perform still image-based face recogni-

tion for each tracked face and combine the recognition results (using combining rules, for example, maximum cumulative probability or a majority vote). The disadvantages of such approaches are the randomness of the combining rules.

In recent years, some researchers try to make use of temporal and spatial information in the video. Zhou et al. (2003) proposed a Bayesian framework based face recognition and tracking which attempts to resolve uncertainties in tracking and recognition simultaneously. A time series state space model, which characterizes the kinematics using a motion vector and the identity using an identity variable, is employed to fuse temporal information. The joint posterior distribution of the motion vector and the identity variable is estimated at each time instant and then propagates to the next time instant. Marginalization over the state vector yields a robust estimate of the posterior distribution of the identity variable. The **sequential importance sampling (SIS)** algorithm was used to estimate the posterior distribution. SIS approximates the posterior density function by a set of random particles with associated weights. Experimental results show the effectiveness of the algorithm.

Li et al. (2001, 2002) applied facial features tracking and face tracking for verification. The basic idea is that if the input is a true face (corresponding to the identity in the database), the tracking trajectories of the facial features or face appearance are basically the same. The corresponding mathematical model is that the distribution of motion vector will have a peak when the face is a true input. Otherwise, the input is an imposter. SIS is also applied to the posterior probability distribution of state variables. However, the estimated probability density needs a large number of particles to characterize the distribution. As a result, the complexity of the algorithm is increased.

"Video - Video" Face Recognition

"Video - Video" face recognition refers to the cases in which both the input and database are shots of video. Based on the use of the information in the video, the existing literatures have the following description methods to represent a shot of video:

1. A vector (corresponding to one frame of the video).
2. A matrix (corresponding to all frames of the video).
3. Probability Density Function (PDF).
4. Dynamic model.
5. Manifold.

Based on the above description methods, "Video – Video" face recognition becomes matching of different description methods. Table 2 shows all possible similarity measures (distances) between two description methods. In Table 2, d stands for the distance or similarity of two models. $f(X)$ stands for probability addition, $M(X)$ stands for majority voting, $D(X)$ stands for the posterior probability.

Some representative methods of "Video - Video" face recognition are briefly introduced below. Torres et al (2002) created a person specific **Principal Component Analysis (PCA)** subspace for each face in the database. The residual distance between the face in one frame and the PCA subspace is used as similarity measure for video indexing. McKenna et al (1997) employed **Gaussian Mixture Model (GMM)** in the reduced PCA subspace to describe each face class. The posterior probability of each face in each frame is computed and the cumulative probability is used as similarity measure. Yamaguchi et al (1998) established PCA subspace for both the input and database video. The distance between the two subspaces is determined by the angle between two subspaces. To further handle the change of illumination, gestures, facial expressions, etc., Fukui & Yamaguchi (2003) further proposed the constraint subspace that includes only the effective component for recognition.

Arandjelovi et al (2005a) used GMM to learn the face distribution. The basis of the approach is the semi-parametric estimation of probability densities confined to intrinsically low-dimensional, but highly nonlinear face manifolds embedded in the high dimensional image space (Arandjelović, Shakhnarovich, Fisher, Cipolla & Darrell, 2004). The **Kullback-Leibler divergence** is adopted as the similarity measure.

Zhou et al. (2003) used the probabilistic model described in previous section. An exemplar-based learning is adopted to automatically select video representatives. The exemplar index is also employed as the state vector. The joint probability density distribution is estimated by sequential importance sampling. Finally, the identity variable is calculated by marginalization. Liu and Chen (2003) proposed a video-based face recognition algorithm based on **Hidden Markov Model (HMM)** which incorporates both the temporal and spatial information. Lee et al. (2003, 2005) approximated face manifolds by a finite number of linear subspaces and used temporal information to robustly estimate the dynamics of the linear subspaces.

Li et al (2001a, 2001b) employed the manifold to represent a shot of video. A 3D shape model is built from 2D images, a shape-and-pose-free textures model and an affine geometrical model. Then, **Kernel Discriminant Analysis (KDA)** is performed to extract the non-linear discriminating features. The identify surfaces are then constructed from these discriminating features. Face recognition is performed by computing **trajectory distance** between the input and database video trajectories.

Table 2. Similarity measures (distance) between two description methods

Database Input	Vector(x)	Matrix(X)	Probability(f)	Dynamic Model(D)	Manifold(M)
Vector(x)	$d(x, x)$	$d(x, X)$	$f(x)$	$D(x)$	$M(x)$
Matrix(X)	$d(X, x)$	d(X, X)	$f(X)$	$D(X)$	$M(X)$
Probability(f)	$f(x)$	$f(X)$	$d(f, f)$	\	\
Dynamic Model(D)	$D(x)$	$D(X)$	\	$d(D, D)$	$d(D, M)$
Manifold(M)	$M(x)$	$M(X)$	\	$d(M, D)$	$d(M, M)$

Table 3. Typical algorithms for "Video - Video" face recognition

Authors	Input Description	Database Description	Measure
Torres et al	Vector(x)	PCA subspace(X)	residual error, $d(x, X)$
McKenna et al	Matrix(X)	GMM(f)	cumulative probability, $f(X)$
Yamaguchi et al	Matrix(X)	PCA subspace(X)	angle distance, $d(X, X)$
Arandjelovi et al	PDF(f)	GMM(f)	K-L divergence, $d(f, f)$
Zhou et al	Dynamic Model(D)	Exemplars(X)	posterior probability, $D(X)$
Liu et al	Dynamic Model(D)	HMM(D)	posterior probability, $d(D, D)$
Lee et al.	Dynamic Model(D)	Finite number of linear subspaces(M)	posterior probability, $d(D, M)$
Li et al	Manifold(M)	Manifold(M)	trajectory distance, $d(M, M)$

Some characteristics of the above reviewed algorithms are listed in Table 3.

FUTURE TRENDS

Video-based face recognition has been actively studied in recent years. How to better exploit both spatial and temporal information in the video sequence is the focus point.

An individual face manifold under various changes (such as expression, pose, illumination, etc) is non-convex and nonlinear. Effective features which can discriminative different classes and tolerate within-class variations are the key for both still image-based and video-based face recognition.

Another trend is to generate a 3D face model from video. See (Zhang, Liu, Dennis, Cohen, Hanson, & Shan, 2004) for an example. The 3D face model can overcome the problem caused by large change of pose and illumination. However, the complexity of 3D model is high.

CONCLUSION

In this article, based on the classification of different scenarios of face recognition methods, Four groups of techniques—the "Image - Image(s)" face recognition, "Image - Video" face recognition, "Video - Image(s)" face recognition, "Video - Video" face recognition are reviewed. Most existing methods of video-based face recognition are surveyed. Their respective advantages and disadvantages are also provided. Some trends of video-based face recognition are summarized. In the future, the approaches will be further investigated to drive more applications.

REFERENCES

Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). Face description with local binary patterns: application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (28)12, 2037-2041.

Arandjelović, O., Shakhnarovich, G., Fisher, G., Cipolla, R., & Darrell, T. (2005a). Face recognition with image sets using manifold density divergence. In: *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition,* San Diego, USA, 581-588.

Arandjelović, O., & Zisserman, A. (2005b). Automatic face recognition for film character retrieval in feature-length films In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, 860-867.

Belhumeur, P.N., Hespanha, J.P., & Kriegman, D. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence. 19 (7), 711–720.

Blanz, V., & Vetter, T. (2003). Face recognition based on fitting a 3D morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (25)9, 1063-1074.

Everingham, M.R., & Zisserman, A. (2004). Automated person identification in video. In: *Proceedings of the 3rd International Conference on Image and Video Retrieval,* Dublin, Ireland, 289-298.

Fukui, K., & Yamaguchi, O. (2003). Face recognition using multi-viewpoint patterns for robot vision. *International Symposium of Robotics Research*, Siena, Italy, 192-201.

Knight, B., & Johnston, A. (1997). The role of movement in face recognition. *Visual Cognition*, 4, 265-274.

Lee, K.C., Ho, J., Yang, M.H., & Kriegman, D. (2003). Video-based face recognition using Probabilistic Appearance Manifolds. In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Madison, USA, 313-320.

Lee, K.C., Ho, J., Yang, M.H., & Kriegman, D. (2005). Visual tracking and recognition using probabilistic appearance manifolds. *Computer Vision and Image Understanding*, (99)3, 303-331.

Li, B., & Chellappa, R. (2001). Face verification through tracking facial features. *Journal of the Optical Society of America A*, 18(12), 2969-2981.

Li, B., & Chellappa, R. (2002). A generic approach to simultaneous tracking and verification in video. *IEEE Transactions on Image Processing*, 11(5), 530-554.

Li, S.Z., & Jain, A.K. (2005). *Handbook of face recognition*. New York: Springer Books.

Li, Y., Gong, S., & Lidell, H. (2001a). Video-based online face recognition using identity surfaces. In: *Proceedings of ICCV Workshop on Recognition, Analysis and tracking of Faces and Gestures in Real-Time Systems*, Vancouver, Canada, 40-46.

Li, Y., Gong, S., & Lidell, H. (2001b). Modeling faces dynamically across views and over time. In: *Proceedings of the IEEE International Conference on Computer Vision,* Vancouver, Canada, 554-559.

Liu, X.M., & Chen, T. (2003). Video-based face recognition using adaptive hidden Markov models. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Madison, USA, 340-345.

McKenna, S., Gong, S., & Raja, Y. (1997). Face recognition in dynamic scenes. In: *Proceeding of British Machine Vision Conference*, (pp. 141-150). Colchester, UK.

Moghaddam, B., Jebara, T., & Pentland, A. (2000). Bayesian face recognition. *Pattern Recognition*, (33)11, 1771-1782.

O'Toole A.J., Roark, D.A., & Abdi, H. (2002). Recognizing moving faces: a psychological and neural synthesis. *Trends in Cognitive Sciences*, (6)6, 261-266.

Phillips, P.J., Grother, P.J., Micheals, R.J., Blackburn, D.M., Tabassi, E., & Bone, J. M. (2003). Face recognition vendor test 2002: evaluation report. *NISTIR* 6965.

Shah, M. (2002). Guest introduction: the changing shape of computer vision in the twenty-first century. *International Journal of Computer Vision*, (50)2, 103-110.

Torres, L. (2004). Is there any hope for face recognition? In: *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Service, Lisbon, Portugal,* 21-23.

Torres, L., & Vila, J. (2002). Automatic face recognition for video indexing applications. *Pattern Recognition*, (35)3, 615-625.

Turk, M., & Pentland, A. (1991). Eigenface for recognition. *Journal of Cognitive Neuroscience*, (3)1, 71-86.

Wiskott, L., Fellous, J.M., Krueuger, N., & Malsburg C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 775-779.

Yamaguchi, O., Fukui, K., & Maeda, K. (1998). Face recognition using temporal image sequence. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, 318-323.

Zhang, Z.Y., Liu, Z.C., Adler, D., Cohen, M.F., Hanson, E., & Shan, Y. (2004) .Robust and rapid generation of animated faces from video images: a model-based modeling approach. *International Journal of Computer Vision*, (58)2, 93-119.

Zhao, W., Chellappa, R., Rosenfeld, A., & Phillips, P. J. (2003). Face recognition: a literature survey. *ACM Computation Survey*, (35)4, 399-458.

Zhou, S., Krueger, V., & Chellappa, R. (2003). Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, (91)12, 214-245.

KEY TERMS

Biometric Authentication: Technologies rely on physical characteristics that are unique for each person to ascertain the identity of an individual.

Face Detection: A computer technology that determines the locations and sizes of human faces in digital images.

Face Recognition: Given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces.

Face Tracking: A computer technology that determines the continuous location of the face(s) on each frame of the image sequence.

Human Face Based Video Retrieval: A process that one searches the video sequences to find the face shot according to the query face image.

Particle Filters: Techniques which also known as Sequential Monte Carlo methods (SMC), are sophisticated model estimation techniques based on simulation.

Sequential Importance Sampling: A very common particle filter algorithm that approximates the probability density functions by a set of random samples with associated weights.

Video-Based Face Recognition: Given a video containing face(s), identify or verify one or more persons using a stored database.

S

Stationary Density of Stochastic Search Processes

Arturo Berrones
Universidad Autónoma de Nuevo León, Mexico

Dexmont Peña
Universidad Autónoma de Nuevo León, Mexico

Ricardo Sánchez
Universidad Autónoma de Nuevo León, Mexico

INTRODUCTION

The optimization of a cost function which has a number of local minima is a relevant subject in many important fields. For instance, the determination of the weights of learning machines depends in general on the solution of global optimization tasks (Haykin, 1999). A feature shared by almost all of the most common deterministic and stochastic algorithms for continuous non – linear optimization is that their performance is strongly affected by their starting conditions. Depending on the algorithm, the correct selection of an initial point or set of points have direct consequences on the efficiency, or even on the possibility to find the global minima. Of course, adequate selection of seeds implies prior knowledge on the structure of the optimization task. In the absence of prior information, a natural choice is to draw seeds from a uniform density defined over the search space. Knowledge on the problem can be gained through the exploration of this space.

In this contribution is presented a method to estimate probability densities that describe the asymptotic behavior of general stochastic search processes over continuously differentiable cost functions. The relevance of such densities is that they give a description of the residence times over the different regions of the search space, after an infinitely long exploration. The preferred regions are those which minimize the cost globally, which is reflected in the asymptotic densities. In first instance, the resulting densities can be used to draw populations of points that are consistent with the global properties of the associated optimization tasks.

BACKGROUND

Stochastic strategies for optimization are essential to most of the heuristic techniques used to deal with complex, unstructured global optimization problems (Pardalos, 2004). The roots of such methods can be traced back to the Metropolis algorithm (Metropolis, Rosenbluth, Rosenbluth, Teller & Teller, 1953), introduced in the early days of scientific computing to simulate the evolution of a physical system to thermal equilibrium. This process is the base of the simulated annealing technique (Kirkpatrick, Gellat & Vecchi, 1983), which makes use of the convergence to a global minimum in configurational energy observed in physical systems at thermal equilibrium as the temperature goes to zero.

The method presented in this contribution is rooted in similar physical principles as those on which simulated annealing type algorithms are based. However, in contrast with other approaches (Suykens, Verrelst & Vandewalle, 1998) (Gidas, 1995) (Parpas, Rustem & Pistikopoulos, 2006), the proposed method considers a density of points instead of Markov transitions of individual points. The technique is based in the interplay between Langevin and Fokker – Planck frameworks for stochastic processes, which is well known in the study of out of equilibrium physical systems (Risken, 1984) (Van Kampen, 1992). Fokker - Planck equation has been already proposed for its application in search algorithms, in several contexts. For instance, it has been used to directly study the convergence of populations of points to global minima (Suykens, Verrelst & Vandewalle, 1998), as a tool to demonstrate the convergence of simulated annealing type algorithms (Parpas, Rustem & Pistikopoulos, 2006) (Geman & Hwang, 1986), or

as a theoretical framework for Boltzmann type learning machines (Movellan & McClelland, 1993) (Kosmatopoulos & Christodoulou, 1994). In the context of global optimization by populations of points, it has been proposed that the populations evolve under the time – dependent version of the Fokker – Planck equation, following a schedule for the reduction of the diffusion constant D (Suykens, Verrelst & Vandewalle, 1998).

In our approach, the stationary version of the Fokker – Planck equation is used to learn the long – term probability density of a general stochastic search process. This is achieved using linear operations and a relatively small number of evaluations of the given cost function.

STATIONARY DENSITY ESTIMATION ALGORITHM

Consider the minimization of a cost function of the form $V(x_1, x_2, ..., x_n, ..., x_N)$ with a search space defined over $L_{1,n} \leq x_n \leq L_{2,n}$. A stochastic search process for this problem is modeled by

$$\frac{dx_n}{dt} = -\frac{\partial V}{\partial x_n} + e(t)$$

(1)

where $e(t)$ is an additive noise with zero mean. Equation (1), known as Langevin equation in the Statistical Physics literature (Risken, 1984) (Van Kampen, 1992), captures the basic properties of a general stochastic search strategy. Under an uncorrelated Gaussian noise with constant strength, Eq. (1) represents a search by diffusion, while a noise strength that is slowly varying in time gives a simulated annealing process. Notice that choosing an external noise of infinite amplitude, the dynamical influence of the cost function over the exploration process is lost, leading to a blind search. The model given by Eq. (1) can be interpreted as a nonlinear dynamical system composed by N interacting particles. The temporal evolution of the probability density of such a system is described by a linear differential equation, the Fokker – Planck equation (Risken, 1984) (Van Kampen, 1992),

$$\frac{dp}{dt} = \frac{\partial}{\partial x}\left[\frac{\partial V}{\partial x}\right] + D\frac{\partial^2 p}{\partial x^2}$$

(2)

The approach proposed in this article is based on the notion of an infinitely long exploration of the search space. In the present model setup for the search, the process converges to a state described by the stationary solution of Eq. (2) (Berrones, 2007). The form of this solution is of the well known Boltzmann type (Risken, 1984) (Van Kampen, 1992). For optimization or deviate generation purposes, its direct use would imply a high computational cost. Instead, a form of Gibbs sampling is proposed in order to estimate the marginal probability density $p(x_n)$ (the details of the following discussion can be consulted in (Berrones, 2007)). The one dimensional projection of Eq. (2) at $t \rightarrow \infty$ leads to the following equation for the conditional cumulative distribution, $y(x_n \mid \{x_j \neq x_n\})$

$$\frac{d^2 y}{dx_n^2} + \frac{1}{D}\frac{\partial V}{\partial x_n}\frac{dy}{dx_n} = 0$$

$$y(L_{1,n}) = 0, \qquad y(L_{2,n}) = 1$$

(3)

Therefore, the estimation of the analytical form of $y(x_n \mid \{x_j \neq x_n\})$ can be achieved by the substitution of the expansion

$$y = \sum_{l=1}^{L} a\varphi_l(x_n)$$

(4)

into Eq. (3). The distribution obtained in this way can be used to draw points from the conditional density $p(x_n \mid \{x_j \neq x_n\})$. According to the principles of Gibbs sampling (Geman & Geman, 1984), the iteration of the previous steps over the N variables will produce a population sampled from the corresponding marginal densities $p(x_n)$. However, in our setup all the information needed to characterize the densities is contained in the coefficients of the expansion (4). In this way, the stationary marginal densities associated to the N variables of the optimization problem, are learned through the averages of the coefficients over the iteration of the random deviate generation process. We call this basic procedure a Stationary Density Estimation Algorithm (SDEA). We have also named the method Stationary Fokker – Planck Machine (SFPM) in (Berrones, 2007), in order to indicate its relation with other methods (Suykens, Verrelst & Vandewalle, 1998) that make use of the Fokker – Planck equation to learn statistical features of stochastic search processes. However, in

(Suykens, Verrelst & Vandewalle, 1998) the Fokker – Planck equation is used to study the evolution of finite populations of points from out of equilibrium states. This contrast with our approach, which estimate the equilibrium densities on the entire search space.

As an example, the SFPM algorithm is tested on the Levy No. 5 function, an important benchmark problem with about *760* local minima and one global optimum (Parsopoulos & Vrahatis, 2002),

$$f(x) = \sum_{i=1}^{5} i \cos((i-1)x_1 + i) \sum_{j=1}^{5} j \cos((j+1)x_2 + j) +$$

$$(x_1 + 1.42513)^2 + (x_2 + 0.80032)^2$$

(5)

with a search space given by the hypercube *[-10, 10]*. The direct implementation of a stochastic search through Eq. (1) would imply the simulation of a stochastic dynamical system composed by two particles with highly nonlinear interactions. By our methodology, in contrast, we are able to obtain adequate densities by linear operations and performing a moderate number of evaluations of the cost function. In Fig.1 the densities generated by *10* iterations of the estimation algorithm with parameters *L=50* and *D=200* are shown. The obtained densities are perfectly consistent with the global properties of the problem, since the known global optimum at the point *(-1.3068, -1.4248)* is contained in the regions with highest probability. The computational

effort is low in the sense of the required number of cost function evaluations, given by *2(L-1)MN=1960*. This is comparable to the effort needed by advanced techniques based on populations in order to obtain good quality solutions for the same problem (Parsopoulos & Vrahatis, 2002). Our approach, however, is not limited to the convergence to good solutions, but it estimates entire densities. The implications of this in, for instance, the definition of probabilistic optimality criteria, are currently under research by us.

FUTURE TRENDS

In our opinion the theory and results presented so far have the potential of considerably enrich the tools for global optimization. The characterization of optimization problems in terms of reliable probability densities may open the door to new insights into global optimization by the use of probabilistic and information – theoretic concepts. From a more practical standpoint, the proposed methodology may be implemented in a variety of ways in order to improve existing or construct new optimization algorithms.

CONCLUSION

This work presents a methodology to estimate the probability density function of optimization problems with

Figure 1. Probability densities, $p(x_1)$ and $p(x_2)$ respectively, generated by 10 iterations of the stationary density estimation algorithm for the Levy No. 5 function. The parameters of the algorithm are L = 50 and D = 200. The global optimum is in the region of maximum probability.

S
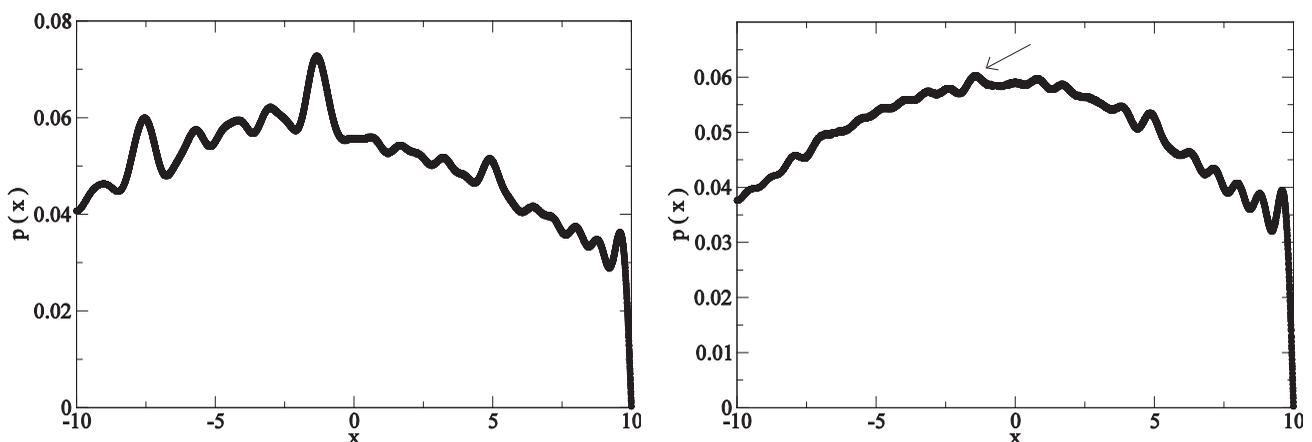
a continuous differentiable cost function, using linear operations and a moderate number of evaluations of the cost function. The generalization to constrained problems appears to be straightforward. This is expected taking into account that the proposed method makes use of linear operations only. In this way, constraints may enter into Eq. (1) as additional nonlinear terms, with no essential increment in computational cost. For instance, combinations of sigmoidal functions can be used for the representation of the constraints as forces produced by energy barriers.

REFERENCES

Berrones, A. (2007). *Generating Random Deviates Consistent with the Long Term Behavior of Stochastic Search Processes in Global Optimization.* Lecture Notes in Computer Science 4507, 1-7, Springer.

Geman, S. & Geman, D. (1984). *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images.* IEEE Transactions on Pattern Analysis and Machine Intelligence (6), 721-741.

Geman, S. & Hwang, C. R. (1986). *Diffusions for Global Optimization.* SIAM Journal on Control and Optimization (24), 5 1031-1043.

Gidas, B. (1995). *Metropolis--type Monte Carlo Simulation Algorithms and Simulated Annealing. In: Topics in Contemporary Probability and its Applications, Prob. Stochastic Ser.*, CRC 159-232.

Haykin, S. (1999). *Neural Networks: a Comprehensive Foundation.* Prentice Hall, New Jersey.

Kirkpatrick, S., Gelatt, Jr., C. D. & Vecchi M. P. (1983). *Optimization by Simulated Annealing.* Science (220) 671-680.

Kosmatopoulos, E. B. & Christodoulou, M. A. The Boltzmann g—RHONN. (1994): *a Learning Machine for Estimating Unknown Probability Distributions.* Neural Networks (7), 2 271-278.

Metropolis, N., Rosenbluth A., Rosenbluth M., Teller A. & Teller E. (1953). *Equations of State Calculations by Fast Computing Machines.* Journal of Chemical Physics (21) 1087-1092.

Movellan, J. R. & McClelland, J. L (1993). *Learning Continuous Probability Distributions with Symmetric Diffusion Networks.* Cognitive Science (17) 463-496.

Pardalos, P. M. & Schoen F. (2004). *Recent Advances and Trends in Global Optimization: Deterministic and Stochastic Methods.* Proceedings of the Sixth International Conference on Foundations of Computer—Aided Process Design, DSI 1-2004 119-131.

Parpas, P., Rustem, B. & Pistikopoulos, E. N. (2006). *Linearly Constrained Global Optimization and Stochastic Differential Equations.* Journal of Global Optimization, (36), 2 191-217.

Parsopoulos, K. E. & Vrahatis, M. N. (2002). *Recent approaches to global optimization problems through Particle Swarm Optimization.* Natural Computing (1) 235-306.

Risken, H. (1984). *The Fokker--Planck Equation.* Springer, Berlin.

Suykens, J. A. K., Verrelst, H. & Vandewalle, J. (1998). *On – Line Learning Fokker—Planck Machine.* Neural Processing Letters, (7), 2 81-89.

Van Kampen, N. G. (1992). *Stochastic Processes in Physics and Chemistry.* North-Holland, Amsterdam.

KEY TERMS

Configurational Energy: Refers to the potential energy associated with the various forces within the elements of a physical system.

Diffusion Constant: Measures the degree of randomness in a diffusion process. The diffusion constant is proportional to the mean square distance moved by particles under diffusion in a given time interval.

Diffusion Process: Random displacement of particles in a physical system due to the action of a temperature.

Gibbs Sampler: A procedure to sample the marginal densities from a high dimensional distribution using one dimensional conditional probabilities.

Heuristic: Is any algorithm that finds a good quality solution to a problem in a reasonable run time.

Learning Machine: This term refers to the development of techniques for automatic extraction of patterns from massive data sets and to the construction of deductive rules. In the context of this article, this concept deals with the automatic learning of densities in global optimization problems.

Random Deviate Generation Process: A process which generates random numbers according to a specific probability distribution.

Search Space: This is the set of all the feasible solutions for an optimization problem.

Stochastic Search: Is an optimization algorithm which incorporate randomness in its exploration of the search space.

Thermal Equilibrium: State in which a physical system is described by probability measures that are time independent.

Statistical Modelling of Highly Inflective Languages

Mirjam Sepesy Maučec
University of Maribor, Slovenia

Zdravko Kačič
University of Maribor, Slovenia

INTRODUCTION

A language model is a description of language. Although grammar has been the prevalent tool in modelling language for a long time, interest has recently shifted towards statistical modelling. This chapter refers to speech recognition experiments, although statistical language models are applicable over a wide-range of applications: machine translation, information retrieval, etc.

Statistical modelling attempts to estimate the frequency of word sequences. If a sequence of words is $s = w_1 w_2 ... w_k$, the probability can be expressed as:

$$P(s) = P(w_1 w_2 ... w_k) =$$

$$\prod_{i=1}^{k} P(w_i | w_1 ... w_{i-1}) \approx \prod_{i=1}^{k} P(w_i | w_{i-n+1} ... w_{i-1}).$$

It is reasonable to simplify this computation by approximating the word sequence generation as a *(n-1)-* order Markov process (Jelinek, 1998). Bigram *(n=2)* and trigram *(n=3)* models are common choices. Although we have limited the context, such models have a vast number of probabilities that need to be estimated. The text available for building the model is called the ´training corpus´ and, typically contains many millions of words. Unfortunately, even in a very large training corpus, many of the possible *n*-grams are never encountered. This problem is addressed by smoothing techniques (Chen & Goodman, 1996).

Which is the best modelling unit? Words are a common choice, but units smaller (or larger) than words can also be used. Word-based *n*-gram is best suited to modelling the English language (Jelinek, 1998). Inflective languages have several characteristics, which harm the prediction powers of standard models.

In general, all Indo-European languages are inflective but a serious problem arises regarding languages which are inflected to a greater extent (e.g. Russian, Czech, Slovenian). Agglutinative languages (e.g. Hungarian, Finnish, Estonian) have even more complex inflectional grammar where, besides inflections, compound words are a big problem. Inflective languages add inflectional morphemes to words. Inflectional morphemes indicate the grammatical information of a word (for example case, number, person, etc.). Inflectional morphemes are commonly added by affixing, which includes prefixing (adding a morpheme before the base), suffixing (adding it after the base), and much less common, infixing (adding it inside the base). A high degree of affixation contributes to the explosion of different word forms, making it difficult, even impossible, to robustly estimate language model probabilities. Rich morphology leads to high OOV (Out-Of-Vocabulary) rates and, therefore, data sparsity is the main problem.

This chapter focuses on modelling unit choice for inflective languages with the aim of reducing data sparsity. Linguistic and data-driven approaches were analyzed for this purpose.

BACKGROUND

Class-Based Language Models

Some words are similar in their morphological, syntactic or semantic functions. In class-based language models, similar words are grouped into classes in order to improve the robustness of parameter estimation:

$$P(w_i | w_{i-1}) = P(w_i | C(w_i)) \cdot P(C(w_i) | C(w_{i-1})).$$

C denotes the deterministic mapping of words into classes. Non-deterministic mapping can also be derived at, where one word can belong to many classes. A model is also applicable, where the word is directly conditioned by the classes of previous words. The idea behind class-based models is parameter-set reduction. There are far fewer free parameters to estimate in a class-based model than in a word-based model.

Words in the same class are similar in a certain way. This similarity can be defined, based on certain external knowledge or statistical criterion. The best known example of clustering using linguistic knowledge is clustering by POS (Part Of Speech). Eight POSs are defined in traditional English grammar: noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection. This set of classes is, however, too small for modelling inflective languages. Those classes that reflect additional grammatical features (gender, case, number, tense, etc.) are more suitable.

Linguistic classes were examined for several languages, which are more or less inflective. A language model for French combined POS classes with a component based on lemmas (El-Beze & Derouault, 1990). In the language model for Czech, words were clustered into 410 morpho-syntactic classes (Nouza & Nouza, 2004). 1300 classes were used in another experiment for Czech (Kolar, Svec & Psutka, 2004). Class-based models with linguistic classes also proved to be successful for Spanish (Casillas, Varona & Torres, 2004).

Data driven classes are automatically derived at by statistical means. IBM pioneered this approach (Brown, de Souza, Mercer, Della Pietra & Lai, 1992). In their approach, words are clustered using a greedy algorithm that tries to minimize the loss of mutual information between classes incurred during the merge. The number of classes must be defined in advance. The algorithm continues to merge pairs of classes until the desired number of classes has been obtained. Another greedy approach uses the exchange algorithm (Martin, Liermann & Ney, 1995). Each word is moved from its class to another one if it maximizes mutual information between classes.

Data-driven class-based language models have been built for many inflective languages. For French they show improved performance on small and large corpora (Zitouni, 2002). The results have been improved by using a hierarchical language model with variable-length class sequences, based on 233 grammatical classes. In experiments on the Russian language, the best results were obtained by using 500 classes (Whittaker & Woodland, 2003). The results were further improved when a class-based model was combined with a word-based model.

Lots of data must be available to derive at classes automatically from the data instead of using external knowledge sources.

Language Models Based on Sub-Word Units

Given the difficulties in language modelling based on full word forms it would be desirable to find a method of decomposing word forms into their morphological components and to build a more robust language model based on probabilities involving individual morphological components.

Lexicons exist for some languages which contain information about the morphological components of words. In experiments on Czech, words were decomposed into stems and endings using a Czech Morphological Analyzer, and were then used as modelling units (Byrne, Hajič, Ircing, Krbec & Psutka, 2000). Morpheme-based language models were also studied for the Korean language, where a word-phrase is an agglomerate of morphemes (Kwon & Park, 2003). Sub-word units are also used when modelling agglutinative languages where, besides inflections, compound words are very common (Szarvas & Furui, 2003). Morphological sub-word units have also been proved for Turkish (Erdoğan, Büyük & Oflazer, 2005). The language model's constraints were represented by a weighted finite state machine.

Many languages do not have developed morphological analysers. Data-driven discovery of a language's morphology is used in such cases. It is common for data-driven approaches to outperform linguistic ones. Morphemic suffixes were discovered by Minimum Description Length (MDL) analysis (Brent, Murthy & Lundberg, 1995). MDL analysis has been used for morphological segmentation for various European languages (Goldsmith, 2001). An algorithm for learning morphology using latent semantic analysis was also discovered (Schone & Jurafski, 2000). This algorithm only extracts affixes when the stem and stem-affix are sufficiently similar semantically. The language model for Russian also improves when using data-driven sub-word units (Whittaker & Woodland, 2000). Language-independent algorithms for discovering word

fragments based on MDL have been presented for Finnish language (Hirsimäki, Creutz, Siivola, Kurimo, Virpioja, & Pylkkönen, 2005). The authors report that word fragments obtained using grammatical rules gave worse results than fragments obtained by data-driven algorithms. They improved speech recognition results furthermore by clustering morph *n*-gram histories (Virpioja & Kurimo, 2006). The same kinds of comparisons with similar conclusions have also been done for the Turkish and Estonian languages.

LANGUAGE MODEL OF INFLECTIVE LANGUAGE

Our work is mainly devoted to the highly inflective Slovenian language. It is a South Slavic language. It shares its characteristics, in varying degrees, with many other inflective languages, especially Slavic.

As in the case of other inflective languages, we concentrate on reducing the perceived data sparsity. The techniques we investigate are language-independent and, as such, also applicable to other highly inflective languages.

Class-Based Language Models of the Slovenian Language

In our first study, the use of data-driven classes was examined. In (Sepesy Maučec, Brest, Kačič & Žumer, 2000) we described an improved algorithm for word clustering. The main idea was to replace the systematic replacement of words between classes with a randomized one. Secondly, instead of replacing one word after another, a randomly selected group of words was replaced at once. The pseudocode of the algorithm is:

1. Set up initial mapping
2. Compute initial train set perplexity PP
3. <u>while</u> (<u>not</u> stopping criterion is met) <u>do</u> <u>begin</u>
4.   randomly select a set of words
5.   for each selected word randomly select target class
6.   compute the new train set perplexity PP1
7.   <u>if</u> (PP1 < PP)
           keep words in new classes <u>and</u> PP:=PP1
    <u>else</u> keep words in old classes
8.   <u>goto</u> step 3
     end

The main bottleneck for a clustering algorithm is time complexity. We developed a parallelized version of the algorithm in order to speed it up. Using random selection, we achieved a 3.7% improvement in perplexity when comparing the results with the basic clustering algorithm, which replaces words systematically.

Having *V* words in the vocabulary and clustered into *C* classes, the space complexity of the class-based bigram language model is $O(C^2 + V)$, in contrast to space complexity $O(V^2)$ of the word-based language model. Using classes, we can enlarge the vocabulary of words by keeping the language model's size small, but this does not solve the problem of OOV words. On the other hand, most speech recognizers use only word-based models. In such cases, class-based models must be converted into word-based ones, which considerably increases the size.

Language Models of Slovenian Language Based on Data-Driven Sub-Word Units

Slovenian words often have many morphological units in common. Two constituent parts can be determined when a highly simplified model of a word is examined: a stem, which can be thought of as responsible for the nuclear meaning of a word, and an ending, which determines the grammatical features. Not all words can be decomposed into stem and ending. In this case an empty ending is used.

In (Sepesy Maučec, Kačič & Horvat, 2004) we showed that it makes sense to model the semantic and grammatical features of words separately:

$$P(w_i | h_i) = P(s_i e_i | h_i) = P(s_i | h_i^*) P(e_i | h_i^{**})$$

w_i is decomposed into a stem s_i and an ending e_i. h denotes previously observed units in the prediction of a stem and an ending.

The prediction of a stem was exposed to topic adaptation. It was presumed that the language in the target environment (where final application would be used) is topically homogeneous. A general language model was tuned to the specific topic by using data at three semantic levels. The first level corresponds to the general language, characterised by the whole corpus. The second level corresponds to the language

characterised by a subset of similar documents. The third level represents a finer level of language topic similarity.

By considering the lengths of histories in predictions, we investigated the following trigram model:

$$P(w_i \mid w_{i-2} w_{i-1}) =$$

$$P\left(s_i \mid s_{i-2} s_{i-1}\right) \cdot \left(\lambda P\left(e_i \mid e_{i-2} e_{i-1}\right) + (1-\lambda) P\left(e_i \mid s_i\right)\right)$$

Prediction of the stem is based on knowledge of the two preceding stems. Prediction of the ending is based on the knowledge of the two preceding endings, and the current stem. In our experiments, the best results were obtained when $\lambda = 0.1$ because a relatively small set of endings can be appended to a particular stem. Some information about word-ending is also contained in the endings of neighbouring words.

The model presupposes a decomposed training corpus. In (Sepesy Maučec, Rotovnik, & Zemljak Jontes, 2003) we used a simple decomposition scheme, based on a preselected set of endings and the longest-match principle. A set of endings was automatically generated over three steps. First, a list was created of all words written in reversed character order. Words were arranged in alphabetical order; thus, words sharing a common ending appear together on the list. The initial characters of adjacent words in the list are compared in order to find a match. Two restrictions were used to avoid over-stemming: the remaining stem should be of a predefined minimum length and the first character of a match must be a vowel. Words should be decomposed at consonant-vowel pair because consonants carry more information about the meaning of word than vowels.

We further improved the decomposition of words in an iterative manner. We searched for the decomposition, which yields the maximized log-likelihood of the training corpus, computed based on sub-word trigrams. The pseudocode of the algorithm is:

1. Collect word bigram counts in train set
2. Set up the initial decomposition
3. Compute the initial log-likelihood of the train set LL
4. while (not stopping criterion is met) do begin
5. randomly select a set of words
6. for each selected word randomly set the new stem-ending boundary
7. compute the new log-likelihood of the train set LL1
8. if (LL1 > LL)

accept new decompositions and LL:=LL1
   else keep old decompositions
9. goto step 4
      end

The choice of initial decomposition is very important, because final decompositions are only guaranteed to be locally optimal. The initial decomposition was set at the decomposition proposed in (Sepesy Maučec et al., 2003). The stopping criterion was a predefined number of iterations.

Experiments have been performed using a newspaper corpus named ´Večer´. The size of the corpus was 85M words (734k distinct words). 14M word bigram counts were collected from the corpus. After initialization, we had 267k distinct sub-words (264k stems and 3k endings) and the initial sub-word perplexity was 361. After 10,000 iterations the number of distinct sub-units increased to 497k (417k stems and 80k endings) but sub-word perplexity decreased to 291. Data-driven decompositions obtained by this algorithm have already been tested in speech recognition experiments (Rotovnik, Sepesy Maučec & Kačič, 2006). The error rate decreased by 6.3% when compared with the results of speech recognition using word-based models.

FUTURE TRENDS

A lot of work has been done on modelling highly inflective languages but there still exists a lack of knowledge on how to model them ´most effectively´. As an extension of the conventional *n*-gram language model, a factored language model has been proposed and tested on Arabic (Bilmes & Kirchhoff, 2003). This factored form could also be useful for other highly inflective languages, because it combines information of different types in one general model. To our knowledge, factored language models have not been widely studied on other highly inflective languages yet, except for Arabic and, more recently, Estonian (Alumae, 2006).

CONCLUSION

This chapter gives an overview of applied methods when modelling highly inflective languages. Considering the characteristics of highly inflective languages we exposed models of two types: class-based and

sub-word based. The motivation behind both of them is data-sparsity reduction.

The main idea of class-based models is to reduce the number of free parameters by clustering words into classes. It is interesting that data-driven classes outperformed linguistic classes in many research experiments.

Sub-word based models reduce the size of the vocabulary by splitting words into smaller units and storing these sub-word units (instead of words) in the vocabulary. Data-driven methods to split words into sub-words surpassed grammatical decompositions for many languages.

The reported experiments regarding the use of these types of models (especially in combination with standard word-based) show an overall reduction of errors in the target applications. We draw the same conclusions from our experiments on the Slovenian language. A promising direction for further work is seen in the factored language model.

REFERENCES

Alumae, T. (2006). Sentence-Adapted Factored Language Model for Transcribing Estonian Speech, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1, 429–432. Toulouse, France.

Bilmes, J., & Kirchhoff, K. (2003). Factored Language Models and Generalized Parallel Backoff, *Proceedings of the Human Language Technology Conference,* 2, 4–6. Edmonton, Canada.

Brent, M., Murthy, S.K., & Lundberg, A. (1995). Discovering Morphemic Suffixes: a Case Study in MDL Induction. *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, 482–490. Fort Lauderdale, Florida.

Brown, P.F., de Souza, P.V., Mercer, R.L., Della Pietra, V.J., & Lai, J.C. (1992). Class-Based N-gram Models of Natural Language, *Computational Linguistics*, 18(4), 467–479.

Byrne, W., Hajič, J., Ircing, P., Krbec, P., & Psutka, J. (2000). Morpheme Based Language Model for Speech Recognition of Czech. *Lecture Notes in Artificial Intelligence*, 1902, 211–216.

Casillas, A., Varona, A., & Torres I. (2003). Experiments with Linguistic Categories for Language Model Optimization. *Lecture Notes in Computer Science*, 2588, 511–515.

Chen, S.F., & Goodman, J. (1996). An Empirical Study of Smoothing Techniques for Language Modelling, *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 310–318. Santa Cruz, California.

El-Beze, M., & Derouault A.M. (1990). A Morphological Model for Large Vocabulary Speech Recognition, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 577–580. Albuquerque, New Mexico.

Erdoğan, H., Büyük, O., & Oflazer, K. (2005). Incorporating Language Constraints in Sub-word Based Speech Recognition. *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 93–103. San Juan, Puerto Rico.

Goldsmith, J. (2001). Unsupervised Learning of Morphology of Natural Language. *Computational Linguistics*, 27(2), 153–189.

Hirsimäki, T., Creutz, M., Siivola, V., Kurimo, M., Virpioja, S., & Pylkkönen, J. (2006). Unlimited Vocabulary Speech Recognition with Morph Language Models Applied to Finnish. *Computer, Speech & Language*, 20(4), 515–541.

Jelinek, F. (1998). *Statistical methods for Speech Recognition*. Cambridge, Massachusetts: MIT Press.

Kolar, J., Svec, J., & Psutka, J. (2004). Automatic Punctuation Annotation in Czech Broadcast News Speech. *Proceedings of the International Workshop on Speech and Computer,* 319–325. Patras, Greece.

Kwon, O.W., & Park, J. (2003). Korean Large Vocabulary Continuous Speech Recognition with Morpheme-based Recognition Units. *Speech Communication*, 39(3-4), 287–300.

Martin, S., Liermann, J., & Ney, H. (1995). Algorithms for Bigram and Trigram Clustering. *Proceedings of the International Conference Eurospeech*, 1253–1256. Madrid, Spain.

Nouza, J., & Nouza, T. (2004). A Voice Dictation System for a Million-Word Czech Vocabulary. *Proceed-*

S

ings of the International Conference on Computing, Communications and Control Technologies, 149–152. Austin, USA.

Rotovnik, T., Sepesy Maučec, M., & Kačič, Z. (2006). Large Vocabulary Continuous Speech Recognition of Inflectional Language with Stems and Endings, *Speech Communication*, 49(6), 437-452.

Schone, P., & Jurafsky, D. (2000). Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. *Conference on Computational Natural Language Learning*, 67–72. Lisbon, Portugal.

Schwenk, H. (2007). Continuous Space Language Models. *Computer, Speech & Language*, 21(3), 492–518.

Sepesy Maučec, M., Brest, J., Kačič, Z., & Žumer, V. (2000). On Solving Statistical Language Modeling for Speech Recognition using a Heterogeneous Computing system (in Slovene). *Electrotechnical Reviews*, 67(1), 55–61.

Sepesy Maučec, M., Rotovnik, T., & Zemljak Jontes, M. (2003). Modelling Highly Inflected Slovenian Language. *International Journal of Speech Technology*, 6(3), 245–257.

Sepesy Maučec, M., Kačič, Z., & Horvat, B. (2004). Modelling Highly Inflected Languages. *Information Sciences*, 166(1-4), 249–269.

Szarvas, M. & Furui, S. (2003). Evaluation of the Stochastic Morphosyntactic Language Model on a One Million Word Hungarian Task. *Proceedings of the International Conference Eurospeech,* 2297–2300. Geneva, Switzerland.

Virpioja, S., & Kurimo, M. (2006). Compact N-gram Models by Incremental Growing and Clustering of Histories. *Proceedings of the International Conference Interspeech*, September 17-21, Pittsburgh, USA.

Whittaker, E.W.D., & Woodland, P.C. (2000). Particle-based Language Modelling. *Proceedings of the International Conference on Spoken Language Processing*, 1, 170–173. Beijing, China.

Whittaker, E.W.D., & Woodland, P.C. (2003). Language Modelling for Russian and English Using Words and Classes. *Computer, Speech & Language*, 17(1), 87–104.

Zitouni, I. (2002). A Hierarchical Language Model Based on Variable-length Class Sequences: The MC [v][n] Approach. *IEEE Transactions on Speech and Audio Processing*, 10(3), 193–198.

KEY TERMS

Corpus: A large collection of texts, usually in electronic form. The corpus has greater value if it is tokenized (segmented into sentences, words etc.) and linguistically annotated (for example POS-tagged and lemmatized).

Inflective Language: A language characterized by the use of inflections. Inflection is the modification of a word in order to reflect grammatical information, such as gender, number, person etc.

Language Model: A description of language. In statistical language modelling it is a set of probability estimates.

***n*-Gram Model**: A model, based on the statistical properties of *n*-grams. *N*-gram model predicts the i-th unit based on the knowledge of *n-1* previous units. In *n*-gram modelling the assumption is made, that each unit depends only on *n-1* previously observed units. This is the main deficiency of *n*-gram modelling, because it has been shown that the range of dependencies is significantly longer.

Out-Of-Vocabulary Rate: Number of unknown words in a new sample of language (it is called a test set), usually expressed in percentage.

Perplexity: A measure of a language model's quality. It can be interpreted as the geometric mean of the branch out factor of the language model. A language model with perplexity X has the same difficulty as an imaginary language in which every word can be followed by X different words with equal probability.

Sub-Word Unit: Modelling unit smaller than a word. Sub-word units are usually morphemes, stems and endings, roots, etc.

Unknown Word: Vocabularies are typically fixed to be tens of thousands of words. All words not in the vocabulary are mapped to a single distinguished word, usually called the unknown word.

Vocabulary: A set of words (or other units) being modelled. The same vocabulary is used by the language model and the target application.

Statistical Simulations on Perceptron–Based Adders

Snorre Aunet
University of Oslo, Norway & Centers for Neural Inspired Nano Architectures, Norway

Hans Kristian Otnes Berge
University of Oslo, Norway

INTRODUCTION

In this article we compare a number of full-adder (1-bit addition) cells regarding minimum supply voltage and yield, when taking statistical simulations into account. According to the ITRS Roadmap two of the most important challenges for future nanoelectronics design are reducing *power consumption* and increasing *manufacturability* (ITRS, 2005).

We use subthreshold CMOS, which is regarded by many as the most promising ultra low power circuit technique. It is also shown that a minimum redundancy-factor as low as 2 is sufficient to make circuits maintain full functionality under the presence of defects. This is, to our knowledge, the lowest redundancy reported for comparable circuits, and builds on a method suggested a few years ago (Aunet & Hartmann, 2003).

A standard Full-Adder (FA) and an FA based on perceptrons exploiting the "mirrored gate", implemented in a standard 90 nm CMOS technology, are shown not to withstand statistical mismatch and process variations for supply voltages below 150 mV. Exploiting a redundancy scheme tolerating "open" faults, with gate-level redundancy and shorted outputs, shows that the same two FAs might produce adequate Sum and Carry outputs at the presence of a defect PMOS for supply voltages above 150 mV, for a redundancy factor of 2 (Aunet & Otnes Berge, 2007).

Two additional perceptrons do not tolerate the process variations, according to simulations. Simulations suggest that the standard FA has the lowest power consumption. Power consumption varies more than an order of magnitude for all subthreshold FAs, due to the statistical variations.

BACKGROUND

The first simple mathematical model of the biological neurons, published by McCulloch and Pitts in 1943, calculates the sign of the weigthed sum of inputs. Sometimes such circuits are called threshold logic gates or threshold elements. Perceptrons may be used to implement Neural Networks as well as digital signal processing.

Nanoscale CMOS technology is expected to be used alongside other technologies in the future. A typical chip will fail if even a single transistor or wire on the chip is defective. Reducing the power consumption and making defect tolerant circuits have been pointed out as important issues (Mead, 1990), (ITRS, 2005).

Reducing the power supply voltage is the most direct and dramatic means of reducing the power consumption (Liu & Svensson, 1993), and subthreshold circuits operating with a supply voltage, V_{dd}, less than the absolute value of the inherent threshold voltages, Vt, has been known for decades (Swensson, Meindl, 1972).

For older technologies, where manufacturability including threshold voltage variability, was not such an important issue (ITRS 2005),(Wong, Mittal, Cao & Starr, 2004) the minimum supply voltages have often been estimated without mismatch and process variations being taken into account (Liu & Svensson, 1993),(Schrom & Selberherr, 1996). To get more realistic estimates we have simulated and compared 4 different topologies for 1-bit addition under statistical variations in the process and matching properties.

MAIN FOCUS OF THE CHAPTER

MOS Transistors in Subthreshold

For an NMOS transistor in subthreshold we have (Andreou, Boahen, Pouliquen, Pavasovic, Jenkins & Strohbehn, 1991):

$$I_{ds,n} = I_0 e^{\frac{\kappa V_{gs}}{V_t}} e^{(1-\kappa)\frac{V_{bs}}{V_t}} \left(1 - e^{-\frac{V_{ds}}{V_t}} + \frac{V_{ds}}{V_0}\right)$$

$I_{ds,n}$ expresses the current from drain to source. I_0 is the zero-bias current where the pre-exponential constants have been absorbed. This includes the channel width ("W") and the length ("L") of the MOSFET structure. V_{gs} is the gate-to-source potential, V_{ds} the drain-to-source potential and Vbs the substrate-to-source potential.

V_0 is the Early voltage, which is proportional to the channel length. κ gives the effectiveness for which the gate potential is controlling the channel current. It is often approximately 0.7-0.75 (Andreou, Boahen, Pouliquen, Pavasovic, Jenkins & Strohbehn, 1991). The thermal voltage is expressed as V_t=kT/q. $V_t = 25.8$ mV at room temperature.

Though equation (1) takes fewer physical effects and nonmonotonous behaviour in certain cases into account, than for example that reported in (Calhoun, Wang & Chandrakasan, 2004), it does provide sufficient insight to make a brief analysis of many subthreshold circuits. A similar equation apply to PMOS transistors, but with opposite polarities.

Experimental Setup for Statistical Simulations of Functionality and Power Consumption for 1-Bit Adders

For statistical (Monte-Carlo) simulations we used a 90 nm standard CMOS process available through CMP (CMP, 2007). Four different Full Adder ("FA") circuits having their inputs driven by inverters, and themselves driving simple inverters were simulated. This is illustrated in figure 1. In the case of no redundancy and faults the lower FA in figure (1) was not included.

For each circuit, at 8 different supply voltages, 100 Monte-Carlo "runs" were done, each having the eight possible combinations of the three inputs, for a total simulated period (transient simulation) of 400 μs, as

illustrated in figure 3 for a case after 5 "runs". This was far from the maximum operational speed of any of the FAs, meaning that the resulting Sum and Carry signals had more than enough time to settle. Each of the 100 runs represented different mismatch and process parameters, and for each run we checked if the circuit was able to produce correct "0" or "1" outputs for all eight input combinations. The yield, shown in figure 4 represents the percentage of the Full Adders (FAs) working for a given supply voltage, out of 100 Monte Carlo "runs".

Redundancy using short circuited driven nodes (Aunet & Hartmann, 2003) was exploited, duplicating each gate for the three FAs based on threshold gates (figure 2). For the other FA only the driven nodes prior to the inverters preceeding the S and C nodes were shorted. A total of 4 PMOS transistors were removed from the 4 FAs (one for each FA), so that each FA missed one PMOS in one of it's threshold gates. This means that each FA in figure 1 had exactly (2N -1) the number of transistors, N, when compared to the previous case with no redundancy.

The average power consumption for the eight input combinations was also calculated. Each of the four circuits perceptrons, with no redundancy, was tested for 8 different supply voltages.

The missing transistor was in the lowermost "min3" gate (figure 2). For the mirrored gate the missing PMOS was the one having the Z input. For the stacked gate as well as the ijcnn gate the missing PMOS was the one between the two other PMOS transistors, referred to figure 2.

For the FA in the upper left corner of figure 2 a PMOS with it's gate connected to the C_{in} input was the one that was removed. Regarding the rest of the setup it was identical to the one in the previous subsection, describing the case without redundancy.

The FAs put to test were a standard CMOS Full Adder containing 28 transistors (upper, left, in figure 2), while the three others were based on the topology in the upper, right, corner of figure 2. They were based on, from left to right in figure 2, the "mirrored gate" (Hampel, Prost & Scheinberg, 1974), the "stacked" gate (Aunet, Berg & Beiu, 2005) and the "ijcnn" gate (Aunet, Oelmann, Abdalla & Berg, 2004), which are all threshold gates.

Regarding transistor dimensions all gate lengths were 100 nm, and all NMOS widths were 220 nm. The standard FA and the "stacked" FA had widths of

all PMOS equal to 400 nm, while the "ijcnn FA" and the "mirrored FA" had PMOS widths of 550 nm and 650 nm, respectively. Buffers, made from two inverters, were inserted on the Sum nodes as well as between the two uppermost threshold gates ("min3") in figure 2.

Results

The percentage of FA circuits that produced correct logic levels for the Sum and Carry signals, under different conditions, are shown in figure 4. It is clear that the standard CMOS FA and the one based on the mirrored gate gives a larger percentage for a given supply voltage when compared to the FAs based on the two other threshold gates.

Power consumption as a function of supply voltage is shown in figure 5, for the basic circuits without any defect transistors or redundancy.

DISCUSSION

The standard Full Adder, and the threshold gate based topology (upper right corner in figure 2) exploiting the mirrored gate, both need supply voltages of at least 150 mV to tolerate mismatch and process variations, according to our simulations. This may be seen to the left in figure 4. The threshold gates "ijcnn" and "stacked" does not tolerate statistical variations like the two previously mentioned solutions, at least not when there are no redundancy and relatively small

Figure 1. Experimental setup for statistical simulation of 1-bit adder
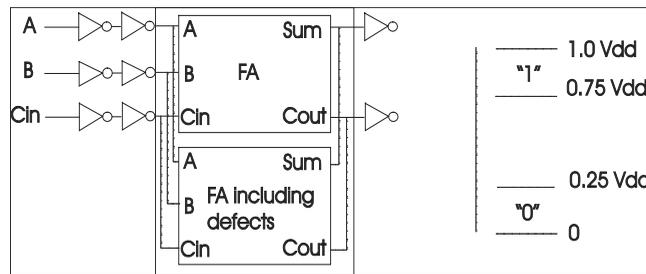
Figure 2. Schematics for the four 1-bit adders (Full Adders). The standard CMOS version is in the upper left corner, while a topology based on perceptrons and inverters is shown in the upper right corner.
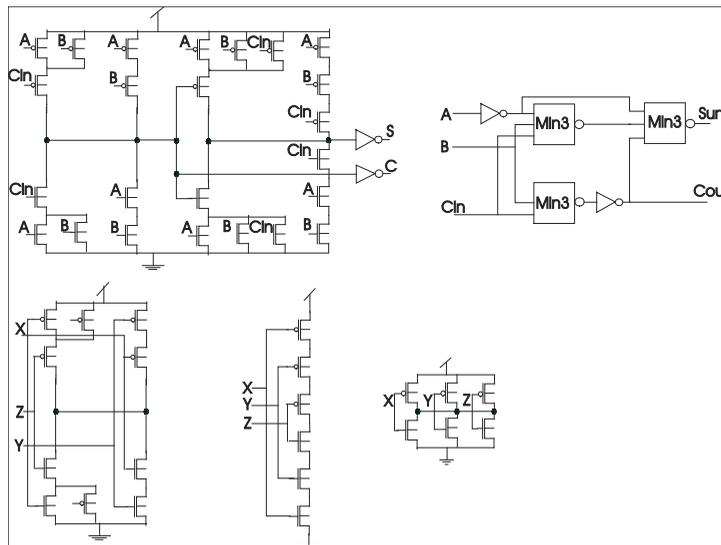
Figure 3. Sum and carry as a function of X, Y and Z inputs for 5 runs
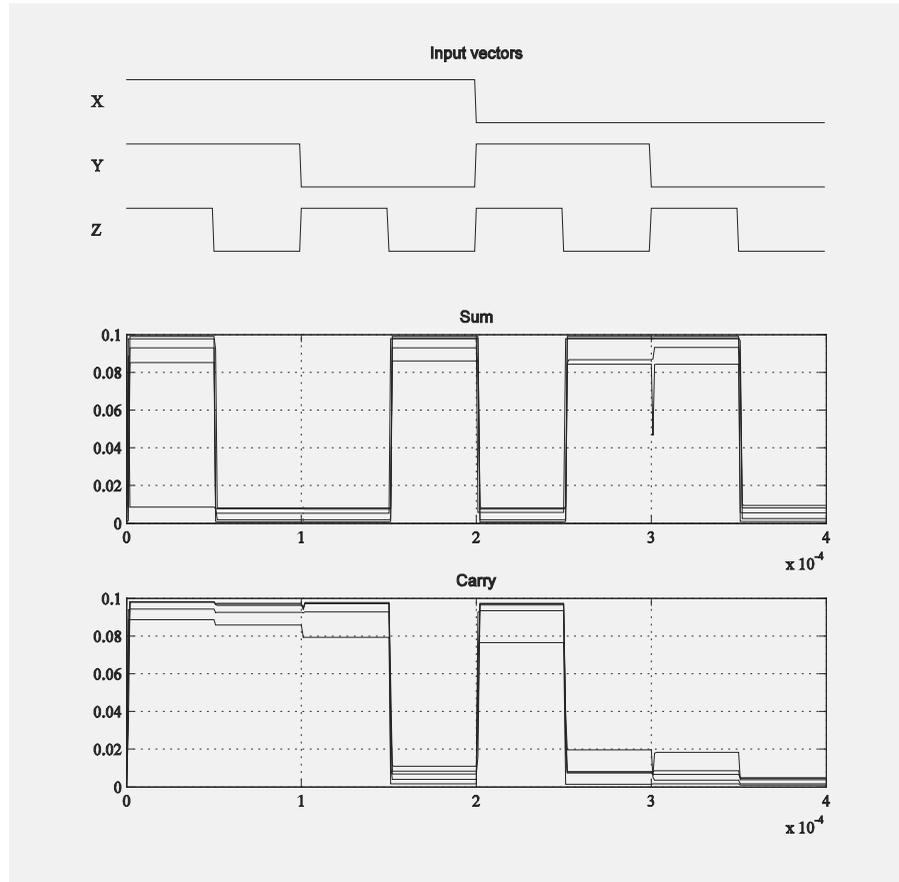
transistors are used. Larger transistors should improve the matching properties and make the circuits less vulnerable to statistical variations in production, as the spread in for example the inherent threshold voltages is inveresly proportional to the square root of the product of the widths and lengths of the MOSFETs (Croon, Decoutere, Sansen & Maes, 2004): $\sigma(V_T) = A_0(V_T) / Sqrt(WL)$.

The mirrored threshold gate was adopted for subthreshold operation and defect-/fault-tolerance using shorted outputs (Aunet & Hartmann, 2003) in (Beiu, Aunet, Nyathi, Rydberg & Djupdal, 2005) and underwent statistical simulations as here, in (Granhaug & Aunet, 2006). Then a redundancy factor of 2 combined with a supply voltage of minimum 175 mV resulted, if a single defect PMOS should be tolerated. In (Granhaug & Aunet, 2006) transistor sizing was slightly different, and the wells of both the PMOS and NMOS transistors

were short circuited, as opposed to our case, where the wells were connected to the rails. For systems of considerable size, implemented in silicon the lowest supply voltage might be 175 mV, reported in (Miyazaki, Kao & Chandrakasan, 2002). Exploiting redundancy, duplicating every gate and tearing one PMOS transistor out from each of the four full-adders gave the results shown to the right in figure 4. The picture is resembling the case to the left, without redundancy, but show some differences. The minimum Vdd to make the standard FA and the one based on the mirrored gate function for all the 100 Monte-Carlo runs was still 150 mV. This is a lower supply voltage than the 175 mV found in (Granhaug & Aunet, 2006). Transistor sizing as well as biasing of wells may have a significant impact on the results, especially in subthreshold, with the many exponential dependencies as shown in equation 1. From figure 4 one can also see that the FAs based on

Figure 4. "Yield" from Monte-Carlo simulations of the FAs at different Vdd's (N=100)
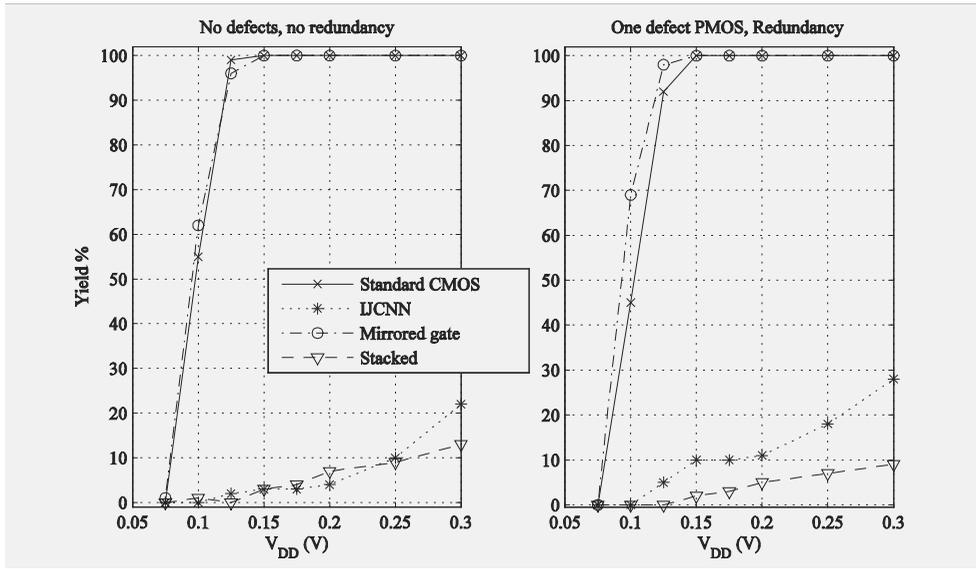
Figure 5. Power consumption for functional FAs without redundancy
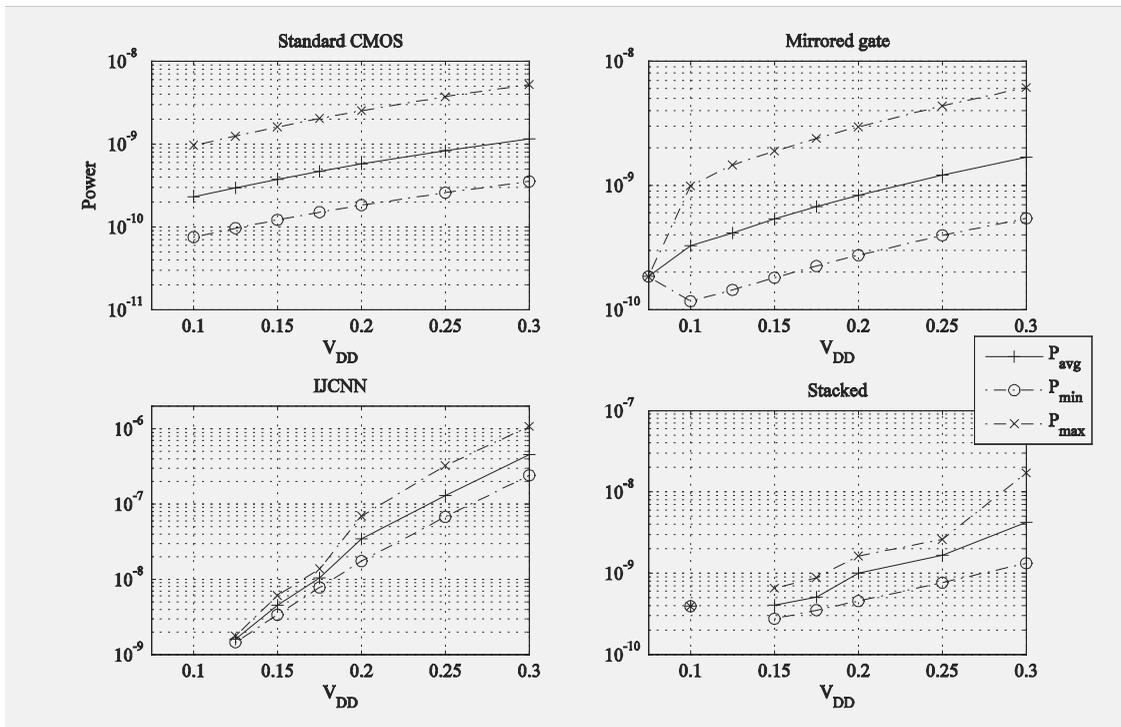
the "mirrored" and the "ijcnn" gates often have a higher yield for a given supply voltage when introducing redundancy and a defect transistor, when compared to the case without redundancy and any defect. The FA based on the "mirrored" gate was the most robust one when there was a defect transistor, giving the highest yield for low supply voltages, according to simulations in figure 4. More simulations, including other defects and additional redundancy could be interesting for future research.

Removing single transistors simulates certain "open" faults in the redundant units, and the scheme using shorted outputs (Aunet & Hartmann, 2003), used in for example (Beiu, Aunet, Nyathi, Rydberg & Djupdal, 2005), (Granhaug & Aunet, 2006) but may not withstand "close" faults like outputs of redundant units shorted to one of the supply rails. A method tolerating such defects as well is presented in (Schmid & Leblebici, 2003). No single technique is enough for tolerating all fault mechanisms in nanoscale circuits and systems, it is concluded in (Lehtonen, Plosila & Isoaho, 2005), so combinations of several methods are needed, depending on the specific design and the proneness to different sources of defects (Lehtonen, Plosila & Isoaho, 2005).

The average, maximum and minium power consumption in the cases where the FAs were able to produce correct logic outputs are shown in figure 5. The standard CMOS FA shows the lowest average power consumption when the supply voltage is above 150 mV, which is when two of the FAs give a "yield" of 100 percent, according to our results. The FA based on the "mirrored" gate shows a slightly higher power consumption, while the FA based on the "ijcnn" gate displays a power consumption up to orders of magnitude above the others, and increasingly so for the relatively higher supply voltages. Even the FAs showing a relatively high tolerance to mismatch and process variations have current levels ranging over more than an order of magnitude, or a factor 10 x, for a given supply voltage. Power consumption for a given supply voltage is expected to increase linearly with the redundancy factor.

The realism in simulations is limited, especially for nanoscale CMOS (Nassif, 2006). So, layout techniques for high matching, including dummy structures , might lead to different results than those presented here.

FUTURE TRENDS

The assumption that a system is composed largely of correctly functioning units is no longer true in emerging nanoelectronics, and reducing the overall power consumption is also among the grand challenges for future nanoelectronics. The low fan-in perceptrons, also called *voters*, or *minority gates*, might be very useful candidates for future nanoelectronics, which has been recently stated (Beiu & Ibrahim, 2007). Defect tolerant subthreshold perceptron circuits exploiting majority gates, as presented here, may thus be useful building blocks for the future.

CONCLUSION

Statistical Monte-Carlo simulations have been performed on 4 Full Adder circuits. For each FA 100 Monte-Carlo runs were done at 8 different subthreshold supply voltages, and the percentage of the runs providing appropriate logic levels for Sum and Carry outputs was calculated. A "yield" of 100 percent meant that a certain FA would tolerate all simulated combinations of statistical variations. The circuits able to reach this limit were a standard FA and an FA based on the "mirrored" threshold gate, both needing a supply voltage, Vdd, above at least 150 mV to guarantee functionality under mismatch and process variations.

When exploiting redundancy and shorting outputs (Aunet & Hartmann, 2003), a supply voltage less than 150 mV is not enough to tolerate the statistical variations when a PMOS is removed from the schematics and a redundancy factor of 2 is used. The standard and mirrored-based FAs are still working for a supply voltage above 150 mV for one defect MOSFET. Power consumption varies by approximately 1 order of magnitude, for all the 4 simulated FAs in subthreshold, with the standard FA having the lowest power consumption at useful supply voltages tolerating large statistical variations.

REFERENCES

Andreou A. G., Boahen K. A., Pouliquen P. O., Pavasovic A., Jenkins R. E., Strohbehn K. (1991), Current-Mode Subthreshold MOS Circuits for Analog

VLSI Neural Systems, *IEEE Transactions on Neural Networks.* 205-213

Aunet S., Berg Y. & Beiu V. (2005), Ultra Low Power Redundant Logic Based on Majority-3 Gates *Proc. IFIP VLSI-SOC,* 553-558

Aunet S. & Hartmann M. (2003), Real-time Reconfigurable Threshold Elements and Some Applications to Neural Hardware. *Proc. 5th International Conference on Evolvable Systems, LNCS.* 365-376

Aunet S., Oelmann B., Abdalla S. & Berg Y. (2004), Reconfigurable subthreshold CMOS perceptron. *Proc. IEEE Int.'l Conf. on Neural Networks,* 1983-1988

Aunet S. & Otnes Berge H. K. (2007), Statistical Simulations for Exploring Defect Tolerance and Power Consumption for 4 1-bit Addition Circuits. *Proc. 9th International Work-Conference on Artificial Neural Networks, LNCS.* 455-462

Beiu V., Aunet S., Nyathi J., Rydberg R. R. III & Djupdal A. (2005), On the advantages of serial architectures for low-power reliable computations. *Proc. IEEE Int.'l Conference on Application Specific Systems, Architectures and Processors*, 276-281.

Beiu V., Ibrahim W. (2007), Why Inverters and Small Fan-In Voters are The Most Promising Gates for Future Nanoelectronics. *Proc. 16th Int.'l Workshop on Post-Binary ULSI Systems, Oslo*.

Calhoun B. H., Wang. A. & Chandrakasan A. (2004), Device sizing for minimum energy operation in sub-threshold circuits. *Proc. Custom Integr. Circ. Conf.* 95-98

CMP ("Circuits Multi Projets") : http://cmp.imag.fr

Croon J. A., Decoutere S., Sansen W. & Maes H. E. (2004), Physical Modeling and Prediction of the Matching Properties of MOSFETs. *Proc. of the European Solid-State Device Research Conference*. 193-196

Granhaug K. & Aunet S. (2006), Improving Yield and Defect Tolerance in Multifunction Subthreshold CMOS Gates. *Proc. 21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems,* 20-28

Hampel D., Prost K. J. & Scheinberg N. R. (1974), *"Threshold Logic using Complementary MOS Device"* U.S. Patent 3900 742, June 24.

(ITRS, 2005) *International Roadmap for Semiconductors,* 2005 Edition - Executive Summary – available: http://www.itrs.net

Lehtonen T., Plosila J. & Isoaho J. (2005), *On Fault Tolerance Techniques towards Nanoscale Circuits and Systems* Turku Center for Comp. Sci., Tech. Rep.

Liu D. & Svensson C. (1993), Trading Speed for Low Power by Choice of Supply and Threshold Voltages, *IEEE Journal of Solid-State Circuits*, 10-17

Mead C. A. (1990), Neuromorphic Electronic Systems, *Proceedings of the IEEE*, 1629-1636

Miyazaki M., Kao J. & Chandrakasan A. P. (2002), A 175 mV Multiply-accumulate unit using an adaptive supply voltage and body bias (asb) architecture. *Proc. IEEE International Solid-State Circuits Conference*, 58-444

Nassif S. R. (2006), Model to Hardware Matching for nano-meter Scale Technologies. *Proc. IEEE Int.'l Conf. on Simulation of Semiconductor Processes and Devices*. 5-8

Schmid A. & Leblebici Y. (2003), Robust Circuit and System Design Methodologies for Nanometer-Scale Devices and Single-Electron Devices. *Proc. Third IEEE Conference on Nanotechnology*. 516-519

Schrom G. & Selberherr S. (1996), Ultra-Low-Power CMOS Technologies. *Proc. International Semiconductor Conference*, 237-245

Swanson R. & Meindl J. D. (1972), Ion-Implanted Complementary MOS Transistors in Low-Voltage Circuits. *Proc. IEEE Int.'l Solid-State Circuits Conf.* 192-193

Wong, B., Mittal, A., Cao Y. & Starr G. W. (2004), *Nano-CMOS Circuit and Physical Design.* ISBN: 978-0-471-46610-9

KEY TERMS

Full Adder: Circuit that produces the binary sum and carry when adding two binary numbers.

Minority-3 Gate: A minority 3 gate outputs a logic "0" signal if, and only if, 2 or 3 out of it's three binary inputs are "1".

Mismatch: Ideally identically constructed elements on an integrated circuits have a mismatch when they differ in their physical properties after production of the chip.

Monte Carlo Simulations: Computer simulations basing the results on statistical distribution of parameters.

Nanoscale CMOS: CMOS technologies where dimensions smaller than 100 nm is critical to the functioning of the produced chip.

Neuron: Electrically excitable cells in the nervous system that process and transmit information.

Parameter Variations: Parameters describing physical traits of integrated circuits may have variations due to mismatch, for example the threshold voltages of transistors.

Perceptron: Type of artificial (feedforward) Neural Network. §

Yield: In this paper the term yield refers to the ratio of functional circuits to the total number of simulated circuits. Often yield refers to the ratio of functional chips to the total number of manufactured chips.

Stochastic Approximation Monte Carlo for MLP Learning

Faming Liang
Texas A&M University, USA

INTRODUCTION

Over the past several decades, multilayer perceptrons (MLPs) have achieved increased popularity among scientists, engineers, and other professionals as tools for knowledge representation. Unfortunately, there is no a universal architecture which is suitable for all problems. Even with the correct architecture, frustrating problems of connection weights training still remain due to the rugged nature of the energy landscape of MLPs. The energy function often refers to the sum-of-square error function for conventional MLPs and the negative log-posterior density function for Bayesian MLPs.

This article presents a Monte Carlo method that can be used for MLP learning. The main focus is on how to apply the method to train connection weights for MLPs. How to apply the method to choose the optimal architecture and to make predictions for future values will also be discussed, but within the Bayesian framework.

BACKGROUND

As known by many researchers, the energy landscape of an MLP is often rugged. The gradient-based training algorithms, such as back-propagation (Rumelhart et al., 1986), conjugate gradient, Newton's method, and the BFGS algorithm (Broyden, 1970, Fletcher, 1970, Goldfarb, 1970, Shanno, 1970), tend to converge to a local minimum near the starting point, rendering the training data learned insufficiently. To reduce the chance of converging to local minima, a number of variants of these algorithms have been proposed based on the idea of perturbation (von Lehmen et al., 1988, Tang et al., 2003 and references therein). In practice, the effects of these perturbations are usually limited, which only delay the learning process converging to local minima a reasonable number of iterations (Ingman & Merlis, 1991).

To avoid the local-trap problem, simulated annealing (SA) (Kirkpatrick et al., 1983) has been employed by some authors to train neural networks. Amato et al. (1991) and Owen & Abunawass (1993) show that for complex learning tasks, SA has a better chance to converge to a global minimum than have the gradient-based algorithms. Geman & Geman (1984) show that the global minimum can be reached by SA with probability 1 if the temperature decreases at a logarithmic rate of O(1/log t), where t denotes the number of iterations. In practice, however, no one can afford to have such a slow cooling schedule. Most frequently, people use a linearly or geometrically decreasing cooling schedule, which can no longer guarantee the global energy minimum to be reached (Holley, et al., 1989).

Other stochastic algorithms that have been used in MLP training include the genetic algorithm (Goldberg, 1989) and Markov chain Monte Carlo (MCMC). Although the genetic algorithm works well for some problems, see, e.g., van Rooij et al. (1996), there is no theory to support its convergence to global minima. MCMC algorithms are mainly used for Bayesian MLPs (MacKay, 1992a, Neal, 1996, Muller & Insua, 1998, de Freitas et al., 2000, Liang, 2003, 2005a,2005b), which will be discussed later.

MAIN FOCUS OF THE CHAPTER

This article presents how the stochastic approximation Monte Carlo (SAMC) (Liang et al., 2007) algorithm can be used for MLP learning, including training, prediction and architecture selection.

A Brief Review for the SAMC Algorithm

Suppose that we are working with the Boltzmann distribution,

$$p(x) = \frac{1}{Z} e^{-U(x)/\tau}, \quad x \in \Omega, \tag{1}$$

where Z is the normalizing constant, $U(x)$ is the energy function, τ is the temperature, and Ω is the sample space. Without loss of generality, we assume that Ω is compact. For MLPs, x denotes the vector of connection weights, and Ω can be restricted to a hyper-rectangle $[-B_\Omega, B_\Omega]^{\dim(\Omega)}$, where B_Ω is a large number such that Ω includes at least a global minimum of $U(x)$. Furthermore, we assume that the sample space can be partitioned according to the energy function into m disjoint subregions: $E_1 = \{x:U(x) \le u_1\}$, $E_2 = \{x:u_1 < U(x) \le u_2\},\ldots, E_{m-1} = \{x:u_{m-2} < U(x) \le u_{m-1}\}$, and $E_m = \{x:U(x) > u_{m-1}\}$, where $u_1,\ldots,u_{m-1}$ are pre-specified real numbers. SAMC seeks to draw samples from each subregion with a pre-specified frequency. If this goal can be achieved, then the local-trap problem can be avoided successfully. Let x_{t+1} denote a sample simulated from the distribution

$$p_{\theta_t}(x) \propto \sum_{i=1}^{m} \frac{\psi(x)}{e^{\theta_{ti}}} I(x \in E_i)$$

(2)

using the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953, Hastings, 1970), where $\Psi(x) = e^{-U(x)/\tau}$ and $\theta_t = (\theta_{t1},\ldots,\theta_{tm})$ is an m-vector in a space Θ. For simplicity, we assume that Θ is compact, e.g., $\Theta = [-B_\Theta, B_\Theta]^{\dim(\Theta)}$ with B_Θ being a large number. Since adding to or subtracting from θ_t a constant will not change $p_{\theta_t}(x)$, θ_t can be kept in the compact set in simulations by adjusting with an additive constant. Let the proposal distribution, $q(x, y)$, of the MH moves satisfy the minorisation condition (Mengersen & Tweedie, 1996), i.e.,

$$\sup_{\theta \in \Omega} \sup_{x,y \in \Omega} \frac{p_\theta(y)}{q(x, y)} < \infty$$

(3)

Since Ω is compact, a sufficient design for the minorisation condition is to choose $q(x, y)$ as a global proposal distribution. A proposal distribution is said global if $q(x, y) > 0$ for all $x, y \in \Omega$. For MLPs, $q(x, y)$ can be chosen as a random walk Gaussian proposal, $y \sim N(x, \sigma^2 I)$, where I is an identity matrix and σ^2 is calibrated such that the MH moves have a desired acceptance rate. As discussed later, restricting the proposal distribution to be global ensures the convergence of the annealing SAMC algorithm to the global energy minima.

Let $\{\gamma_t\}$ be a positive non-decreasing sequence satisfying the conditions:

i. $$\sum_{t=0}^{\infty} \gamma_t = \infty,$$

ii. $$\sum_{t=0}^{\infty} \gamma_t^\delta < \infty$$

for some $\delta \in (1, 2)$. For example, one can set

$$\gamma_t = \left(\frac{t_0}{\max(t_0, t)} \right)^\eta$$

(4)

for some values of $t_0 > 1$ and

$$\eta \in (\frac{1}{2}, 1).$$

A large value of t_0 will allow the sampler to reach all subregions very quickly, even in the presence of multiple local minima. Let $\pi = (\pi_1,\ldots, \pi_m)$ be an m-vector with $0 < \pi_i < 1$ and

$$\sum_{i=1}^{m} \pi_i = 1,$$

which defines a desired sampling frequency distribution on the subregions. With the above notations, an iteration of SAMC can be described as follows.

SAMC Algorithm

a. Generate $x_{t+1} \sim K_{\theta_t}(x_t,.)$ with a single MH step:
   1. Generate y according to the proposal distribution $q(x_t, y)$.
   2. Calculate the ratio

      $$r = e^{\theta_{tJ(x_t)} - \theta_{tJ(y)}} \frac{\psi(y)}{\psi(x_t)} \frac{q(y, x_t)}{q(x_t, y)},$$

      where $J(x)$ denote the index of the subregion that the sample x belongs to.
   3. Accept the proposal with probability min(1, r). If it is accepted, set $x_{t+1} = y$; otherwise, set $x_{t+1} = x_t$.

b. Set $\theta^* = \theta_t + \gamma_t (e_{t+1} - \pi)$, where γ_t is called the gain factor, $e_{t+1} = (e_{t+1,1},..., e_{t+1,m})$, and $e_{t+1,i} = 1$ if $x_{t+1} \in E_i$ and 0 otherwise.

c. If $\theta^* \in \Theta$, set $\theta_{t+1} = \theta^*$; otherwise, set $\theta_{t+1} = \theta^* + c^*$, where c^* is a constant vector and is chosen such that $\theta^* + c^* \in \Theta$. The existence of c^* is obvious, since B_Θ has been set to a large number and it is reasonable to assume that $\max_{i=1}^m \theta_i^* - \min_{i=1}^m \theta_i^* \ll B_\Theta$ holds at each iteration.

A remarkable feature of SAMC is its self-adjusting mechanism. If a proposal is rejected, the weight of the subregion that the current sample belongs to will be adjusted to a larger value, and thus a proposal of jumping out from the current subregion will be less likely rejected in the next iteration. This mechanism effectively prevents the system from getting trapped in local minima. This is very important for MLP training as its energy landscape is often rugged.

SAMC falls into the category of stochastic approximation algorithms (Robbins & Monro, 1951, Andrieu et al., 2005 and references therein). The convergence of SAMC can be extended from a theorem presented in Liang et al. (2007). Under mild conditions and as $t \to \infty$,

$$\theta_{ti} \to \begin{cases} C + \log \left(\int_{E_i} \psi(x)dx \right) - \log(\pi_i + \zeta), & E_i \neq \varnothing, \\ -\infty, & E_i = \varnothing, \end{cases} \quad (5)$$

where

$$\zeta = \sum_{j \in \{i : E_i = \varnothing\}} \pi_j \Big/ (m - m_0)$$

and $m_0 = \#\{i : E_i = \varnothing\}$ is the number of empty subregions, and C is an arbitrary constant. A subregion E_i is said to be empty if

$$\int_{E_i} \psi(x)dx = 0.$$

In SAMC, the sample space partition can be made blindly by simply specifying some values $u_1,..., u_{m-1}$. This may result in some empty subregions. The constant C can be determined by imposing a constraint on θ_t, say,

$$\sum_{i=1}^m e^{\theta_{ti}}$$

is equal to a known number. In addition, Liang (2007) shows that θ_t can converge in the form L_2 at a rate of $O(1/t)$. Let $\hat{\pi}_{ti} = P(x_t \in E_i)$ be the probability of sampling from the subregion E_i at iteration t. Equation implies that as $t \to \infty$, $\hat{\pi}_{ti}$ will converge to $\pi_i + \zeta$ if $E_i \neq \varnothing$ and 0 otherwise. This further implies that as the number of iterations goes to infinity, SAMC can approximately draw samples from each of the subregions with a pre-specified probability. With an appropriate specification of π, sampling can be biased to the low energy regions to increase the chance of finding the global minimum.

Annealing SAMC for MLP Learning

In theory, SAMC is able to find the global energy minima if the run is long enough. However, due to the broadness of the sample space, the process may be slow even when sampling is biased to low energy subregions. To accelerate the search process, one can iteratively shrink the sample space in simulations. As argued below, this modification preserves the theoretical property of SAMC when a global proposal distribution is used.

Suppose that the subregions $E_1,...,E_m$ have been arranged in ascending order by energy; that is, if $i < j$ then $U(x) < U(y)$ for any $x \in E_i$ and $y \in E_j$. Let $\kappa(u)$ denote the index of the subregion that a sample x with energy u belongs to. Let Ω_t denote the sample space at iteration t. Annealing SAMC, which will be abbreviated as ASAMC hereafter, starts with

$$\Omega_1 = \bigcup_{i=1}^m E_i ,$$

and then iteratively sets

$$\Omega_t = \bigcup_{i=1}^{\kappa(U_{\min}^t + \Delta)} E_i \quad (6)$$

where $U_{\min}^t$ is the minimum energy value obtained by iteration t, $\Delta > 0$ is a user specified parameter. The sample space Ω_t shrinks iteration by iteration. In this sense, the modified algorithm is called ASAMC.

Since the proposal distribution is global, the convergence property of SAMC still holds for ASAMC on the limiting space $\Omega_\infty = \lim_{t \to \infty} \Omega_t$, although Ω_∞ may contain some separated regions. The existence of Ω_∞ is true due to the monotonicity of the sequence $\Omega_1 \supseteq$

S

$\Omega_2 \supseteq \cdots$. If follows from Scheffe's theorem (Scheffe, 1947) that as $t \to \infty$, x_t will converge in distribution to a random variable with density

$$p_\theta(x) \propto \sum_{i=1}^{\kappa(u_{min}+\Delta)} \frac{(\pi_i + \zeta)\psi(x)}{\int_{E_i} \psi(x)dx} I(x \in E_i), \qquad (7)$$

where u_{min} denotes the global minimum of the energy function $U(x)$. Again, as in SAMC, the convergence can be attained in the L_2 form at a rate of $O(1/t)$. If we let Δ go to zero, then the ASAMC samples will converge in distribution to the global minima of $U(x)$.

For an effective implementation of ASAMC, several issues need to be considered.

Sample space partitioning. Since within the same subregion, ASAMC is reduced to sampling from the unnormalized density $\Psi(x)$, we suggest that the maximum energy difference in each subregion should be bounded by a reasonable number, say, 2τ, to ensure that the local Metropolis-Hastings moves within the same subregion have a reasonable acceptance rate.

Choice of Δ. The performance of ASAMC depends on the value of Δ to some extent. If Δ is too large, ASAMC may take a long time to locate the global minimum due to the broadness of the sample space. If Δ is too small, ASAMC may also take a long time to locate the global minimum. In this case, the sample space may contain only a few separated regions, and the most proposed transitions will be rejected. In our experience, a value of Δ between 5 and 10 works well for most MLP problems.

Desired sampling distribution. The choice of π is not critical to the efficiency of ASAMC, as in which the sample space has been shrinked with iterations. On the contrary, in SAMC, π should be chosen carefully to bias sampling to low energy regions to improve ergodicity of the simulation.

Gain factor. To estimate the integrals

$$\int_{E_1} \psi(x)dx, \dots, \int_{E_m} \psi(x)dx$$

accurately, γ_t should be very close to 0 at the end of simulations. Otherwise, the resulting estimates may have a large variation. The decreasing speed of γ_t can be controlled by t_0 and η. In practice, we often fix $\eta = 1$ and vary the value of t_0 according to the complexity of the problem. The more complex the problem is, the larger value of t_0 one should choose.

Convergence diagnostic. A formal diagnostic for the convergence of ASAMC should base on multiple runs. A rough diagnostic for a single run can be done by comparing the observed sampling frequencies and the desired sampling frequencies of different subregions. If they match with each other very well, we may regard the run converged. Otherwise, one may re-run the algorithm with a larger number of iterations or a larger value of t_0.

ASAMC has been compared in Liang (2007) with simulated annealing, SAMC, and the BFGS algorithm on a number of examples, including the famous N-parity and two-spiral problems. The numerical results for the two-spiral problem are re-presented in Table 1 and

Table 1. Comparison of ASAMC, SAMC, SA and BFGS for the two-spiral problem. Notations: let z_i denote the minimum energy value obtained in the ith run. "Mean"$=\sum_{i=1}^{20} z_i/20$, "SD" is the standard deviation of "mean", "Minimum"$= \min_{i=1}^{20} z_i$, "Maximum"$= \max_{i=1}^{20} z_i$, "Proportion"$=\#\{i : z_i \leq 0.2\}$, "Iteration" is the average number of iterations performed in each run, and "Time" is the average CPU time cost by each run.

Algorithm	Mean	SD	Minimum	Maximum	Proportion	Iteration(10^6)	**Time**
ASAMC	**0.620**	**0.191**	**0.187**	**3.23**	**15**	**7.1**	94m
SAMC	**2.727**	**0.208**	**1.092**	**4.09**	**0**	**10.0**	132m
SA-1	**17.845**	**0.706**	**9.020**	**22.06**	**0**	**10.0**	123m
SA-2	**6.433**	**0.450**	**3.030**	**11.02**	**0**	**10.0**	123m
BFGS	15.500	0.899	10.00	24.00	0	---	3s

Figure 1. Classification maps learned for the two-spiral problem by ASAMC with a MLP of 30 hidden units. The black and white points show the training data for the two different spirals, respectively. (a) Classification map learned in a run. (b) Classification map averaged over 20 run.
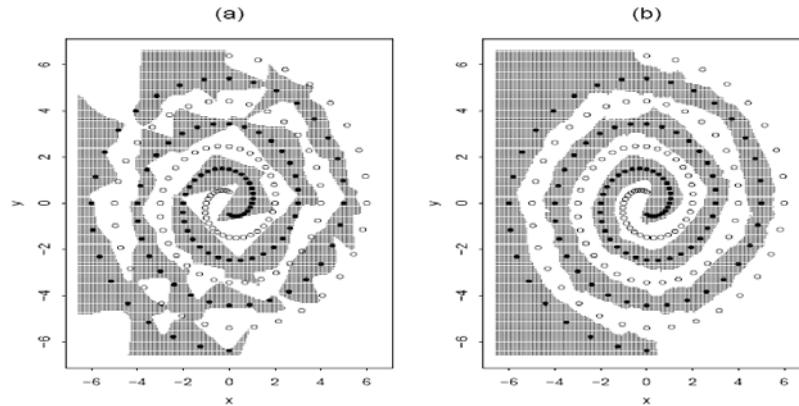
Figure 1. Refer to Liang (2007) for the settings of the respective algorithms. The results for the other examples are similar. In summary, ASAMC outperforms the other algorithms in both training and test errors. Like other stochastic algorithms, ASAMC requires longer training time than do the gradient-based algorithms. It provides, however, an efficient approach to train MLPs for which the energy landscape is rugged.

Bayesian MLP Learning

SAMC can also be used for training Bayesian MLPs. Let $\Psi(x)$ denote the posterior density of a MLP (up to a normalizing constant), and $\widehat{g}_i = \lim_{t \to \infty} e^{\theta_{ti}}$. Thus, the following density

$$\widehat{p(x)} \propto \sum_{i=1}^{m} \frac{\Psi(x)}{\widehat{g}_i} I(x \in E_i) \tag{8}$$

can work as a trial density for sampling from $\Psi(x)$. As a trial density, it possesses two nice properties. First, the importance weight is bounded above by $\max_i \widehat{g}_i$, assuming that $\widehat{g}_i$ has been normalized by an additional constraint, e.g.,

$$\sum_{i=1}^{m} \widehat{g}_i$$

is a known constant. Second, sampling from $\widehat{p(x)}$ will lead to a random walk in the space of nonempty sub-

regions if we regard each subregion as a point. Hence, the whole sample space can be well explored.

Suppose that important samples $(x_1, w_1),...,(x_n, w_n)$ have been drawn from using a MCMC sampler, where w_i denotes the importance weight of x_i. Let $f(z \mid x)$ denote the output of the MLP with input z. For a new input z_0, the Bayesian point prediction is then

$$\widehat{f(z_0)} = \frac{\sum_{i=1}^{n} w_i f(z_0 \mid x_i)}{\sum_{i=1}^{n} w_i} \tag{9}$$

Evidence Evaluation for Bayesian MLPs

In addition to MLP learning, SAMC also provides a convenient way for evaluating evidence of Bayesian MLPs. As pointed out by MacKay (1992b), the Bayesian evidence can be used as a guideline of architecture selection for Bayesian MLPs. Let $f(D|x)$ denote the likelihood function of a given MLP model, and let $l(x)$ denote the prior density imposed on x. As before, we suppose that Ω has been restricted to a compact set. Define the function

$$\Psi(x, k) = \begin{cases} f(D \mid x)l(x), & k = 1 \\ 1/|\Omega|, & k = 0 \end{cases} \tag{10}$$

on the product space $\Omega \times \{0,1\}$, where $|\Omega|$ denotes the hypervolume of the space Ω. Partition the product space as follows: $E_0 = \{(x, k) : k = 0, x \in \Omega\}$, $E_1 = \{(x, k) : k =$

$1, U(x) \leq u_1\}, \cdots, E_m = \{(x, k) : k = 1, U(x) > u_{m-1}\}$. If SAMC is run with this partition, the evidence of the MLP can then be estimated by

$$\widehat{EV} = \frac{\sum_{i=1}^{m}(\pi_i + \zeta)\widehat{g_i}}{(\pi_0 + \zeta)\widehat{g_0}} g_0, \tag{11}$$

where

$$g_0 = \int_{E_0} \psi(x,0)dx,$$

$$\widehat{g_i} = \lim_{t\to\infty} e^{\theta_{ti}},$$

and $0 < \pi_0 < 1$. We note that $\Psi(x,0)$ can be any non-negative function with g_0 being analytically available.

FUTURE TRENDS

In the future, we need to carry out a series of comparisons to assess the ability of SAMC in different aspects. For example, we need to compare SAMC with advanced MCMC samplers, such as parallel tempering (Geyer, 1991) and evolutionary Monte Carlo (Liang & Wong, 2001), to assess its ability in Bayesian prediction; and to compare SAMC with the Gaussian approximation method (MacKay, 1992b) to assess its ability in evidence evaluation.

CONCLUSION

This article proposes an innovative method for MLP training, prediction, and architecture selection. The strength of SAMC comes from its self-adjusting mechanism, which enables it to overcome the local-trap problems. Like simulated annealing and genetic algorithms, SAMC avoids the requirement for the gradient information of the objective function. Hence, it can be used as a general optimization, simulation, and integration tool in many other problems, such as combinational optimization, model selection, and statistical simulations.

REFERENCES

Amato, S., Apolloni, B., Caporali, G., Madesani, U., & Zanaboni, A. (1991). Simulated annealing approach in back-propagation. *Neurocomputing*, 3(5-6), 207-220.

Andrieu, C., Moulines, E., & Priouret, P. (2005). Stability of Stochastic Approximation Under Verifiable Conditions. *SIAM J. Control and Optimization*, 44(1), 283-312.

Broyden, C.G. (1970). The convergence of a class of double rank minimization algorithms. *J. Inst. Maths. Applns*, 6(3), 76-90.

de Freitas, N., Niranjan, M., Gee, A.H., & Doucet, A. (2000). Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4), 955-993.

Fletcher, R. (1970). A new approach to variable metric algorithms. *Computer J.*, 13(3), 317-322.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6), 721-741.

Geyer, C.J., (1991). Markov chain Monte Carlo maximum likelihood, in C*omputing Science and Statistics: Proceedings of the 23rd Symposium on the interface* (E.M. Keramigas, ed.), pp.156-163, Fairfax: Interface Foundation.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, & Machine learning*, Addison Wesley.

Goldfarb, D. (1970). A family of variable metric methods derived by variational means. *Maths. Comp.*, 24(109), 23-26.

Hastings, W.K. (1970). Monte Carlo Sampling Methods Using Markov Chain and Their Applications. *Biometrika*, 57(1), 97-109.

Holley, R.A., Kusuoka, S. & Stroock, D. (1989). Asymptotic of the spectral gap with applications to the theory of simulated annealing. *Journal of Functional Analysis*, 83(2), 333-347.

Ingman, D. & Merlis, Y. (1991). Local minimization escape using thermodynamic properties of neural networks. *Neural Networks*, 4(3), 395-404.

Kirkpatrick, S., Gelatt, C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

Liang, F. (2003). An effective Bayesian neural network classifier with a comparison study to support vector machine. *Neural Computation*, 15(8), 1959-1989.

Liang, F. (2005a). Bayesian neural networks for nonlinear time series forecasting. *Statistics and Computing*, 15(1), 13-29.

Liang, F. (2005b). Evidence evaluation for Bayesian neural networks using contour Monte Carlo. *Neural Computation*, 17(6), 1385-1410.

Liang, F. (2007). Annealing stochastic approximation Monte Carlo algorithm for neural network training. *Machine Learning*, 68(3) 201-233.

Liang, F., Liu, C. & Carroll, R.J. (2007). Stochastic Approximation in Monte Carlo Computation. *Journal of the American Statistical Association*, 102(477), 305-320.

Liang, F. and Wong, W.H. (2001). Real parameter evolutionary Monte Carlo with applications in Bayesian mixture models. *Journal of the American Statistical Association,* 96(454), 653-666.

MacKay, D.J.C. (1992a). A practical Bayesian framework for backprop networks. *Neural Computation*, 4(3), 448-472.

MacKay, D.J.C. (1992b). The evidence framework applied to classification problems. *Neural Computation*, 4(5), 720-736.

Mengersen, K.L. & Tweedie, R.L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24(1), 101-121.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6), 1087-1091.

Muller, P. & Insua, D.R. (1998). Issues in Bayesian analysis of neural network models. *Neural Computation*, 10(3), 749-770.

Neal, R.M. (1996). *Bayesian Learning for Neural Networks*. New York: Springer.

Owen, C.B. & Abunawass, A.M. (1993). Applications of simulated annealing to the back-propagation model improves convergence, *SPIE Proceedings*, 1966, 269-276.

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics,* 22(3), 400-407.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by back-propagating errors. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1* (D.E. Rumelhart and J.L. McClelland, ed.), pp.318-362, Cambridge, MA: MIT Press.

Scheffe, H. (1947). A useful convergence theorem for probability distributions. *Ann. Math. Statist.*, 18(3), 434-438.

Shanno, D.F. (1970). Conditioning of quasi-Newton methods for function minimization. *Maths. Comp.*, 24(111), 647-656.

Tang, Z., Wang, X. Tamura, H., & Ishii, M. (2003). An algorithm of supervised learning for multilayer neural networks. *Neural Computation*, 15(5), 1125-1142.

van Rooij, A.J.F., Jain, L.C., & Johnson, R.P. (1996). *Neural Network Training Using Genetic Algorithms*. Singapore: World Scientific.

von Lehmen, A., Paek, E.G., Liao, P.F., Marrakchi, A., & Patel, J.S. (1988). Factors influencing learning by back-propagation. In *Proceedings of IEEE International Conference on Neural Networks*, pp.335-341, New York: IEEE Press.

KEY TERMS

Genetic Algorithm: A search heuristic used in computing to find true or approximate solutions to global optimization problems.

Markov Chain Monte Carlo (MCMC): A class of algorithms for sampling from probability distributions by simulating a Markov chain that has the desired distribution as its stationary distribution. The state of the Markov chain after a large number of steps is then used as a sample from the desired distribution.

Metropolis-Hastings Algorithm: A popular MCMC algorithm with the acceptance probability $\{1,[f(y)q(y,x)]/[f(x)q(x,y)]\}$ for a new state y given the current state x, where $f(\cdot)$ is the target distribution and $q(\cdot,\cdot)$ is the proposal distribution.

Model Evidence: The log-marginal likelihood of the data obtained by integrating out the parameters over the space of models. Its value expresses the preference shown by the data for different models.

Multiple Layer Perceptron (MLP): An important class of neural networks, which consists of a set of source nodes that constitute the input layer, one or more layers of computational nodes, and an output layer of computational nodes. The input signal propagates through the network in a forward direction, on a layer-by-layer basis.

Simulated Annealing: A generic probabilistic meta-algorithm used to find true or approximate solutions to global optimization problems.

Stochastic Approximation Algorithm: A probabilistic meta-algorithm suggested by Robbins and Monro (1951) for solutions of regression equations.

S

Stream Processing of a Neural Classifier I

M. Martínez-Zarzuela
University of Valladolid, Spain

F. J. Díaz Pernas
University of Valladolid, Spain

D. González Ortega
University of Valladolid, Spain

J. F. Díez Higuera
University of Valladolid, Spain

M. Antón Rodríguez
University of Valladolid, Spain

INTRODUCTION

An *Artificial Neural Network* (ANN) is a computational structure inspired by the study of biological neural processing. Although neurons are considered as very simple computation units, inside the nervous system, an incredible amount of widely inter-connected neurons can process huge amounts of data working in a parallel fashion. There are many different types of ANNs, from relatively simple to very complex, just as there are many theories on how biological neural processing works. However, execution of ANNs is always a heavy computational task. Important kinds of ANNs are those devoted to pattern recognition such as *Multi-Layer Perceptron* (MLP), *Self-Organizing Maps* (SOM) or *Adaptive Resonance Theory* (ART) classifiers (Haykin, 2007).

Traditional implementations of ANNs used by most of scientists have been developed in high level programming languages, so that they could be executed on common *Personal Computers* (PCs). The main drawback of these implementations is that though neural networks are intrinsically parallel systems, simulations are executed on a *Central Processing Unit* (CPU), a processor designed for the execution of sequential programs on a *Single Instruction Single Data* (SISD) basis. As a result, these heavy programs can take hours or even days to process large input data. For applications that require real-time processing, it is possible to develop small ad-hoc neural networks on specific hardware like *Field Programmable Gate Arrays* (FPGAs). However, FPGA-based realization of ANNs is somewhat expensive and involves extra design overheads (Zhu & Sutton, 2003).

Using dedicated hardware to do machine learning was typically expensive; results could not be shared with other researchers and hardware became obsolete within a few years. This situation has changed recently with the popularization of *Graphics Processing Units* (GPUs) as low-cost and high-level programmable hardware platforms. GPUs are being increasingly used for speeding up computations in many research fields following a *Stream Processing Model* (Owens, Luebke, Govindaraju, Harris, Krüger, Lefohn & Purcell, 2007).

This article presents a GPU-based parallel implementation of a Fuzzy ART ANN, which can be used both for training and testing processes. Fuzzy ART is an unsupervised neural classifier capable of incremental learning, widely used in a universe of applications as medical sciences, economics and finance, engineering and computer science. CPU-based implementations of Fuzzy ART lack efficiency and cannot be used for testing purposes in real-time applications. The GPU implementation of Fuzzy ART presented in this article speeds up computations more than 30 times with respect to a CPU-based C/C++ development when executed on an NVIDIA 7800 GT GPU.

BACKGROUND

Biological neural networks are able to learn and adapt its structure based on the external or internal information that flows through the network. Most types of ANNs present the problem of *catastrophic forgetting*. Once the network has been trained, if we want it to learn from new inputs, it is necessary to repeat the whole training process from the beginning. Otherwise, the ANN would forget previously acquired knowledge. S. Grossberg developed the *Adaptive Resonance Theory* (ART) to address this problem (Grossberg, 1987). Fuzzy ART is an extension of the original ART 1 system that incorporates computations from *fuzzy set theory* into the ART network, and thus making it possible to learn and recognize both analog and binary input patterns (Carpenter, Grossberg & Rosen, 1991).

GPUs are being considered in many fields of computation and some researchers have made efforts for integrating different kinds of ANNs on the GPU. Most research has been done for implementing *Multi-Layer Perceptron* (MLP) taking advantage of the GPU performance in matrix-matrix products (Rolfes, 2004) (Oh & Jung 2004) (Steinkraus, Simard & Buck 2005). Other researchers have used the GPU for *Self-Organizing Maps* (SOM) with great results (Luo, Liu & Wu, 2005) (Campbell, Berglund & Streit, 2005). Bernhard et al. achieved a speed increase of between 5 and 20 times simulating large networks of *Spiking Neurons* on the GPU (Bernhard & Keriven, 2006). Finally, Martínez-Zarzuela et al. developed a generic *Fuzzy ART* ANN on the GPU achieving a speed up higher than 30 over a CPU (Martínez-Zarzuela, Díaz, Díez & Antón, 2007).

Commodity graphics cards provide a tremendous computational horsepower. NVIDIA's GeForce 7800 GTX GPU is able to sustain 165 GFLOPS against the 25.6 GFLOPS theoretical peak for the SSE units of a dual-core 3.7 GHz Intel Pentium Extreme (Owens, Luebke, Govindaraju, Harris, Krüger, Lefohn & Purcell, 2007). Newest generation of graphics cards, like NVIDIA Geforce 8800 Ultra, or AMD (ATI) Radeon HD 2900 XT, can give a peak performance higher than 500 Gflops and 100 GB/s peak memory bandwidth. Graphics cards manufacturers have recently discovered the field of high performance computing as to be a target market for their products and are providing specific hardware and software to couple with enterprises and researchers heavy computational requirements.

FUZZY ART NEURAL NETWORK STREAM PROCESSING

This article describes a parallel implementation of a Fuzzy ART ANN using a *stream processing model*. In this uniform parallel processing paradigm a series of computations, defined by one function or *kernel*, are made over an ordered set of data or *stream* on a *Single Instruction Multiple Data* (SIMD) basis. The main restriction of the model is also one of the reasons it can provide large increases in performance and a simplified programming model: operations on each stream element are independent, allowing the execution of the kernel on different hardware processing units simultaneously, and avoiding stalls that could occur because of inter-units data sharing.

GPUs used to have two types of programmable processors, namely *vertex* and *fragment* processors. Both kinds of processors were devised to operate on four component vectors, as the basic primitives of 3D computer graphics are 3D vertices in projected space (*x*, *y*, *z*, *w*) and four component colors (*red*, *green*, *blue*, *alpha*). Both vertex and fragment units could be used to execute a *kernel* over a *stream of data (Stream Processing)* and are programmed using *shaders* that can be written using high level languages as Cg (Randima & Kilgard, 2003), GLSL or HLSL. Latest generation of GPUs, like nVIDIA GeForce 8800 GTX, do not include fragment of vertex processors, but unified *Stream Processors* (SPs): generalized floating-point scalar processors capable of operating on vertices, pixels, or any manner of data. These new GPUs can be programmed using CUDA *(Compute Unified Device Architecture)* Toolkit from nVIDIA. CUDA is a promising new software development solution for programming GPUs, simplifying software development by using the standard C language. Before CUDA was launched programming GPUs for *General Purpose* computation (GPGPU) involved translating algorithms into graphics terms (Harris, 2005). Other companies like *Rapidmind* are developing easy-to-program APIs that use just-in-time (JIT) compilers for translating source code into a format that will work on several system's hardware (GPU, Cell or an x86 CPU). Arrays of data can be uploaded from the CPU to the GPU memory and stored in *textures*. RGBA textures can be used to store 4 floating point data per texture unit *(texel)*. Data is modified along the *graphics pipeline* and then written to the *frame-buffer* memory or rendered to a

S

new texture, allowing a direct feedback of the output to the pipeline's entry.

Fuzzy ART Equations

Fuzzy ART systems are comprised of three layers or fields of nodes. First layer receives the input vector denoted by $\vec{I} = (I_1, \cdots, I_M)$. Nodes in the output layer represent the active code or category of the input pattern being selected. For each output neuron, a *choice function* T_j ($j : 1...N$) is defined by:

$$T_j(\vec{I}) = \frac{|\vec{I} \wedge w_j|}{\alpha + |w_j|},$$

(1)

where $w_j = (w_{j1},...,w_{jM})$ denotes associated *Long-term Memory* (LTM) trace, fuzzy MIN operator $\wedge$ is defined by $(p_i \wedge q_i) \equiv \min(p_i, q_i)$ and the norm $|\cdot|$ is defined by

$$|\vec{p}| \equiv \sum_{i=1}^{M} |p_i|.$$

Category choice is indexed by J, where $T_J = \max(T_j : j = 1...N)$ and system enters in *resonance* if the *match function* meets the *vigilance criterion*:

$$\frac{|\vec{I} \wedge w_J|}{|\vec{I}|} \geq \rho.$$

(2)

When this occurs, vector w_j is updated using (3). Otherwise, node J is inhibited making $T_j = 0$. If no node

j is found to meet the vigilance criterion, a new output neuron is committed.

$$w_J^{new} = \beta(\vec{I} \wedge w_J^{old}) + (1 - \beta)w_J^{old}$$

(3)

Fuzzy ART Training Process on the GPU

Learning is not a parallel but a sequential process. Different input patterns cannot be learned at the same time, because they would all generate different categories. Optimizing the training process for parallel execution must be done when searching for the category that most resembles the input pattern. Fuzzy ART implementations on the CPU sequentially compute the activity for every output node. Then, a sort operation is made in order to know which neuron is most fired by the input pattern (1). If the category stored in this neuron resembles input pattern (2), its associated weights are updated with the new information (3); otherwise, the next most fired neuron must be analyzed. In a *parallel stream processing* implementation, we can compute the choice function for every output neuron (1) in a parallel fashion. Moreover, we can obtain the match function (2) for every node simultaneously.

In a GPU implementation, weights of every committed neuron w_j are stored as rows in a texture W^T. Input pattern $\vec{I}$ is rendered to every row of a texture I^T with same dimensions as W^T, so that during category choice, it can be compared to every LTM traces at once, as it is shown in Fig. 1a). Global operations over the elements of a *stream of data*, such as calculating its maximum or the sum are tricky to perform in a GPU

Figure 1. Training process of a Fuzzy ART ANN on a GPU

(a) Performing $I^T \wedge W^T$

b) Choosing J_{th} neuron and updating w_J
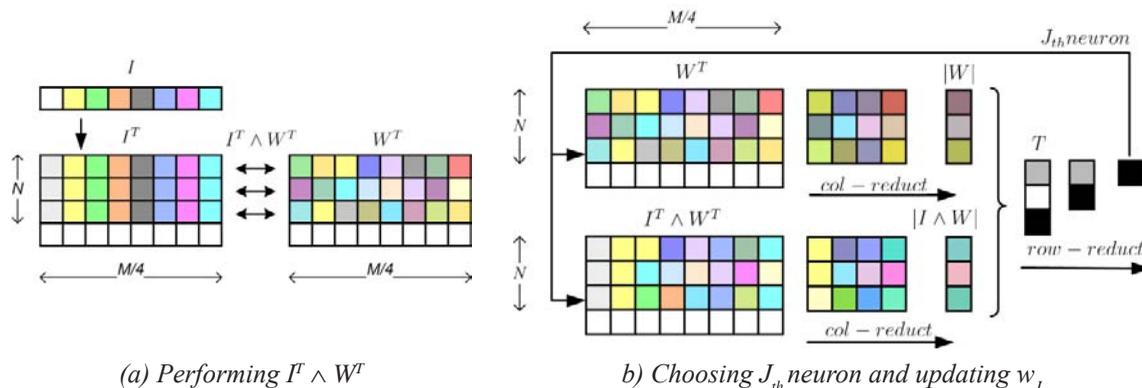
and must be accomplished by doing several render passes. A *ping-pong technique* consists in using the output of a rendering pass as input in the next one. In each pass a local operation is made between neighborhood elements in a texture and the results are written to a smaller texture. After a series of *reductions*, the final result is obtained (Horn, 2005). Calculating the norms $|\vec{I} \wedge w_j|$ and $|w_j|$ is made using a *column reduction operation* along textures W^T and $I^T \wedge W^T$, as it is shown in Fig 1b).

The use of RGBA textures allows running MIN and SUM operations on 4-component vectors in one clock cycle on every *fragment shader* unit, making the process faster. If dimensions of input patterns are not multiple of 4, unused channels of the RGBA textures must be padded with zeros. Reduced textures are then used to store the activity of each neuron, satisfying the match criteria, on the R channel of a texture T; the G channel is used to store the category index; the A channel takes the value of 1 in case the match criteria is satisfied and 0 otherwise; finally, channel B can be used for printing the matching rate, which can be very useful for debugging purposes.

The J_{th} neuron is found using a *row reduction operation* over texture T, in which those fragments not satisfying the match criteria are discarded. If the system enters in resonance, the weights of the selected category are updated by rendering into the corresponding sub-region of texture W^T. If not, the new pattern is learned by rendering to an unused row of weights in W^T according to equation (3).

Fuzzy ART Testing Process on the GPU

The Fuzzy ART testing algorithm is easier and much more profitable to implement on the GPU. In this process, several input patterns can be categorized in a parallel fashion when learning mode is switched off. The best data configuration takes advantage of every *stream processor* available on the GPU for categorizing each pattern in several *shader passes*. Fig. 2 shows the organization of the data on the GPU. In the proposed system, for every (x,y) coordinate pair on the input data, a pattern is stored along the z direction. A single RGBA texture can store 4 component input vectors, and several RGBA textures can be used to store greater patterns. After N *shader passes*, being N the number of committed categories by the network, an output RGBA texture, containing classification information for every pattern, can be obtained. In Fig. 2.b) it is shown the output for *shader pass* 30.

A texture W^T is used for storing F_2 field neuron weights on the GPU. Each row stores a LTM trace w_j, just as in the training implementation. Input vector components stored in RGBA input textures are compared with corresponding column of weights on W^T. In each *shader pass*, the activation of the k*th* output neuron and the match function are computed for every input

Figure 2. Testing process of a fuzzy ART ANN on a GPU

a) Classification between N categories through N shader passes

b) Output classification results after shader pass 30
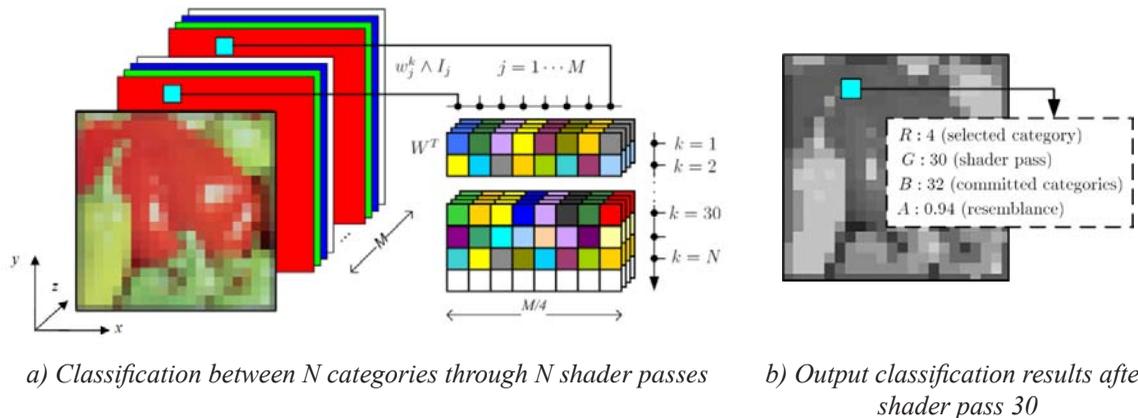
pattern. These values are rendered into an RGBA output texture, which is used as input for the next iteration, again using the *ping-pong* technique. If the activation in pass k is bigger than the computed activation in pass $k-1$ and the match criteria is satisfied, then the index category is updated on the output texture. Rendering both the index of the selected category and the match function to the output texture allows the expert to visually analyze the result: different levels on channel R represent different categories and *alpha* channel shows the level of resemblance of the input pattern to the selected category.

EXPERIMENTAL RESULTS

In order to measure the performance of the implementation, several tests were done on a CPU with a Fuzzy ART C++ self-written implementation and on a GPU using the previously described C++/OpenGL/Cg implementation. Timings were taken on a 3.2 GHz Pentium 4 with 1 GB RAM and a GeForce 7800GT 256 MB.

Performance of Fuzzy ART relays on several factors: length of the input pattern $\vec{I}$, number of input patterns P presented to the network and number of committed categories N. During the learning process, N varies depending both on the grade of similarity between patterns and the *vigilance parameter* ρ (2). For the training tests, a synthetic *benchmark*, comprised of several sets of patterns, was generated. In each set, the length of input vectors M and the number of expected categories N vary (see Table 1). In order to guarantee N was not too influenced by M and P, a *Multivariate Normal Distribution* was used for pattern generation. Being N the number of categories in a set of P patterns $\vec{I}_p = (\vec{a}, \vec{a}^c), p = 1 \cdots P$, the k patterns belonging to category N_i within the set, were generated using a normal distribution for each vector $\vec{a} \sim N_N(\mu, \sum)$, and then obtaining its complement coding $\vec{a}^c$. In vector μ, the mean for every component is selected to be in the $(0,1)$ range and covariances were set to null in covariance matrix Σ. Finally, parameters in the network were chosen to be $\rho = 0.9$, $\alpha = 0.05$ and $\beta = 1$.

Table 1 reveals that the training process takes more time to execute on the GPU than on the CPU. As stated before, learning is a sequential process, thus we cannot re-write Fuzzy ART learning algorithm for an optimal parallel execution. However, the proposed design demonstrated to be faster than a Matlab implementation of Fuzzy ART, where even a collection of 50×10^3 patterns with dimension 4 takes 380 s to train. Performance of training is expected to grow in applications where the number of committed nodes is very large, so that fragment processors are in use for longer periods of time.

For measuring the time taken by the testing process, a different collection of benchmarks was generated and the ANN was tested using previously stored LTM traces.

Table 1. Times for training and testing on a CPU and on a GPU

M	P (x 10³)	N	TRAIN		TEST		
			CPU (s)	GPU (s)	CPU (s)	GPU (s)	SPEEDUP
	10	15	0,0582	4,2128	0,0535	0,0014	38,5
4	50	59	0,4606	25,2468	0,4704	0,0145	32,4
	100	119	1,4212	53,8550	1,4954	0,0563	26,6
	10	8	0,0595	4,7471	0,0545	0,0012	46,2
8	50	50	0,5706	30,8801	0,5919	0,0157	37,8
	100	100	1,8734	65,3028	1,9809	0,0605	32,7
	10	10	0,0743	6,0570	0,0702	0,0018	38,9
16	50	55	0,9131	35,1509	0,9075	0,0300	30,3
	100	111	3,3745	70,2651	3,3425	0,1181	28,3
	10	10	0,0961	6,2251	0,0932	0,0029	32,7
32	50	50	1,4913	35,3596	1,4449	0,0523	27,6
	100	100	5,3758	74,8078	5,3725	0,2135	25,2
MEAN							**33,1**

In this case, the GPU demonstrated to be many times more efficient than the CPU (see Table 1). In the GPU-based testing implementation several input patterns can be categorized in parallel, deeply exploiting the GPU streaming programming model. As it is shown in Table 1, testing process can perform the classification of 32-component patterns between 100 different categories at a rate of 4.68×10^5 patterns per second and classify 4-component patterns between 15 different categories at a rate of 7.14×10^6 patterns per second.

FUTURE TRENDS

Described implementation of Fuzzy ART training algorithm on the GPU is still slower than a high-level programmed implementation on the CPU. In the proposed implementation patterns, which are to be learned by the network, are downloaded from the CPU to the GPU one by one causing GPU to stall, waiting for new data. This represents a serious bottleneck. Furthermore, when the number of committed categories is not very high, *arithmetic intensity* of the design is very low, because there are a limited number of operations that can be made with uploaded data. Future research tasks can include the use of *Pixel Buffer Objects* (PBOs), an OpenGL extension, to achieve fast asynchronous transfer rates from CPU to GPU memory.

CONCLUSION

A GPU implementation of a Fuzzy ART Neural Network following a *stream processing model* was introduced in this paper. This design successfully faces the problem of integrating both training and testing processes on a commodity graphics card following a *stream processing model*.

Fuzzy ART testing process is performed on the GPU up to x46 times faster than in a CPU allowing its use for real-time applications which involve pattern recognition and decision making. Training process, though, is still slower on the GPU than on the CPU.

GPUs are quickly evolving and every 6-9 months a new generation of improved processors is made publicly available. Forward compatibility of the presented implementation for future hardware releases is guaranteed and greater performance can be expected with newer cards.

REFERENCES

Bernhard, F., & Keriven, R. (2006). Spiking neurons on gpus. In Peter M.A. Sloot Vassil N. Alexandrov, Geert Dick van Albada and Jack Dongarra, editors, *Computational Science – ICCS 2006*, *LNCS* 3994, pp. 236–243. Springer.

Campbell, A., Berglund, E., & Streit, A. (2005). Graphics hardware implementation of the parameter-less self-organising map. In *IDEAL*, pp. 343–350.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4(6), pp. 759–771.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11, pp. 23–63.

Harris, M. (2005). Mapping computational concepts to gpus. In Matt Pharr, editor, *GPU Gems 2*, chapter 31, pp. 493–508. Addison Wesley.

Haykin, S. (2007). *Neural Networks: a Comprehensive Foundation (3rd Edition)*. Prentice-Hall, Inc.

Horn, D. (2005). Stream reduction operations. In Matt Pharr, editor, *GPU Gems 2*, chapter 36, pp. 573–587. Addison Wesley.

Lefohn A., Kniss, J., & Owens, J. (2005). Implementing efficient parallel data structures on gpus. In Matt Pharr, editor, *GPU Gems 2*, chapter 33, pp. 521–544. Addison Wesley.

Luo, Z., Liu, H. & Wu, X. (2005). Artificial neural network computation on graphic process unit. In *IJCNN '05: Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, pages 622–626, Montreal, Canada.

Martínez-Zarzuela, M., Díaz, F.J., Díez, J.F. & Antón, M. (2007). Fuzzy ART Neural Network Parallel Computing on the GPU. In F. Sandoval, ed: *International Work-Conference on Artificial Neural Networks (IWANN '07)*, San Sebastián, Spain. Springer LNCS (4507), pp. 463-470.

Oh, K-S. & Jung, K. (2004). GPU implementation of neural networks. *Pattern Recognition*, 37(6), pp. 1311–1314.

S

Owens, J., Luebke D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., & Purcell, T. (2007). A Survey of General-Purpose Computation on Graphics Hardware. *Computer Graphics Forum*, 26(1), pp. 80–113.

Randima, F., & Kilgard, M. (2003). *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*. Addison Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Rolfes, T. (2004). Artificial neural networks on programmable graphics hardware. In *Game Programming Gems 4 (Game Programming Gems Series)*. Charles River Media, Inc., Rockland, MA, USA.

Steinkraus, D., Simard, P.Y., & Buck, I. (2005). Using gpus for machine learning algorithms. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pp. 1115–1119, Washington, DC, USA. IEEE Computer Society.

Zhu, J., Sutton, P. (2003). FPGA implementation of neural networks - a survey of a decade of progress. *Proceedings of the 13th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1062–1066, Lisbon.

KEY TERMS

ART *(Adaptive Resonance Theory)***:** Learning theory developed by S. Grossberg that is used in competitive neural systems and includes short-term-memory (STM) and long-term-memory (LTM) processes.

Fuzzy ART: Evolution of the ART1 neural network capable of learning normalized analog input patterns in an unsupervised way through the use of fuzzy operators.

Fuzzy Logic: Mathematical method originated from the fuzzy set theory, which allows the partial membership of elements in a set, dealing with approximate reasoning instead of exactly deduced from classical logic.

GPGPU *(General-Purpose computation on GPUs)***:** A recent trend in computer science consisting in the use of the *Graphics Processing Unit* (GPU), for doing expensive computational tasks rather than just computer graphics.

GPU *(Graphics Processing Unit)***:** A dedicated graphics rendering device very efficient at manipulating and displaying computer graphics, thanks to its highly parallel structure.

Neural Classifier: An artificial neural network utilized to identify input patterns as members of a predefined class (supervised classification) or as members of an unknown class (unsupervised classification).

Stream Processing: A paradigm for the execution of parallel processing operations exploiting *data-level parallelism* rather than *task-level parallelism* that provides incredible performance with minimal programming effort.

Stream Processing of a Neural Classifier II

M. Martínez-Zarzuela
University of Valladolid, Spain

F. J. Díaz Pernas
University of Valladolid, Spain

D. González Ortega
University of Valladolid, Spain

J. F. Díez Higuera
University of Valladolid, Spain

M. Antón Rodríguez
University of Valladolid, Spain

INTRODUCTION

This article presents a real-time Fuzzy ART neural classifier for skin segmentation implemented on a *Graphics Processing Unit* (GPU). GPUs have evolved into powerful programmable processors, becoming increasingly used in time-dependent research fields such as dynamics simulation, database management, computer vision or image processing. GPUs are designed following a *Stream Processing Model* and each new generation of commodity graphics cards incorporates rather more powerful and flexible GPUs (Owens, 2005).

In the last years *General Purpose GPU* (GPGPU) computing has established as a well-accepted application acceleration technique. The GPGPU phenomenon belongs to larger research areas: *homogeneous and heterogenous multi-core computing*. Research in these fields is driven by factors as the *Moore's Gap*. Today's uni-processors follow a 90/100 rule, where 90 percent of the processor is passive and 10 percent is doing active work. By contrast, multi-core processors try to follow the same general rule but with 10 percent passive and 90 percent active processors when working at full throughput. Single processor *Central Processing Units* (CPUs) were designed for executing general purpose programs comprised of sequential instructions operating on single data. Designers tried to optimize complex control requirements with minimum latency, thus many transistors in the chip are devoted to branch prediction, out of order execution and caching.

In the article *Stream Processing of a Neural Classifier I* several terms and concepts related to GPGPU were introduced. A detailed description of the Fuzzy ART ANN implementation on a commodity graphics card, exploiting the GPU's parallelism and vector capabilities, was given. In this article, the aforementioned Fuzzy ART GPU-designed implementation is configured for robust real-time skin recognition. Both learning and testing processes are done on the GPU using chrominance components in TSL *(Tint, Saturation and Luminance)* color space. The Fuzzy ART ANN implementation recognizes skin tone pixels at a rate of 270 fps on an NVIDIA GF7800GTX GPU.

BACKGROUND

Human body parts detection has important applications as a first step in many high-level computer vision tasks such as personal identification, video indexing systems and *Human-Machine Interfaces* (HMI). HMI needs real-time video processing while consuming as few system resources as possible. Skin color is widely used as a cue for detecting and tracking targets containing skin, such as faces and hands in an image. The final objective of skin color detection is to build a decision rule to segment skin and non-skin pixels in an image efficiently. The simplest solution defines skin colors as those that have a certain range of values in the coordinates of a color space. OpenVidia was one of the first computer-vision oriented developments able to run skin

tone segmentation on the GPU (Fung, 2005). For this purpose OpenVidia uses RGB *(Red, Green and Blue)* to HSV *(Hue, Saturation and Value)* color conversion and threshold filtering.

Statistical approaches for skin segmentation are based on the assumption that skin colors follow a certain distribution which can be estimated. These approaches normally make use of the chrominance components in a color space, thresholds and tunable parameters.

Neural Network approaches have been proposed to learn skin color distribution. Karlekar et al. used a MLP neural network to classify pixels into skin and non-skin colors (Karlekar & Desai, 1999). More complex models have been proposed to deal with changing conditions, such as varying illumination in the images. Sahbi et al. used an ANN for coarse level skin detection, and then the areas found were subjected to Gaussian color modeling with a fuzzy clustering approach (Sahbi & Boujemaa, 2000). Martínez-Zarzuela et al. used a GPU-based Fuzzy ART ANN implementation to learn skin colors in TSL *(Tint, Saturation and Luminance)* color space (Martínez-Zarzuela, Díaz, González, Díez & Antón, 2007). In their system, Fuzzy ART categorization process takes advantage of every fragment processor available in the GPU, so that several pixels can be tested simultaneously by the network, allowing recognition at high frame rates.

Some other researchers have made efforts for integrating different kinds of ANNs on the GPU for speeding up specific applications. Oh et al. developed a GPU-based MLP for text area classification in an image; achieving almost 20 times speed up over a CPU (Oh & Jung, 2004). Luo et al. implemented a MLP on the GPU for real-time ball recognizing and tracking in a soccer robot contest (Luo, Liu & Wu, 2005). Steinkraus et al. proposed using graphics cards for OCR and on-line handwritten recognition (Steinkraus, Simard & Buck, 2005). Finally, Bernhard et al. developed two image segmentation algorithms using spiking neural networks on the GPU (Bernhard & Keriven, 2006).

STREAM PROCESSING FOR ANN-BASED SKIN RECOGNITION

TSL Color Space

Color filtering is a powerful tool in computer vision applications including the detection and tracking of human body parts. Color processing has low computational cost and is robust against geometrical transformations (e.g. rotation, scaling, transfer and shape changes). However, factors such as non-idealities in color cameras and illumination conditions can spoil the performance of filtering-based applications.

Color can be decomposed into three different components, one luminance and two chrominance components. Several researches have proved that skin colors have a certain invariance regarding chrominance components. Skin tone and lighting mainly affect the luminance value (Hsieh, Fan & Lin, 2005).

Different color spaces separating chrominance and luminance components have been used for skin color segmentation: YIQ, YCbCr, CIE-Lab, CIE-Luv, HSV, IHS and TSL (Phung, Bouzerdoum & Chai, 2005). In TSL color space (Terrillon, David & Akamatsu, 1998), a color is specified in terms of *Tint* (T), *Saturation* (S) and *Luminance* (L) values. TSL has been selected as the best color space to extract skin color from complex backgrounds (Duan-sheng & Zheng-kai, 2003) because it has the advantage of extracting a given color robustly while minimizing illumination influence. The equations to obtain the T, S and L components in normalized TSL space are:

$$T = \frac{1}{2\pi} \arctan\left(\frac{r^{'}}{g^{'}}\right) + \frac{1}{2}, \tag{1}$$

$$S = \sqrt{\frac{9}{5}(r^{'2} + g^{'2})}, \tag{2}$$

$$L = 0.299R - 0.587G + 0.114B, \tag{3}$$

where $r' = (r - 1/3)$ and $g' = (g - 1/3)$, being r and g the chrominance components of the normalized rgb color model. The values of T, S and L are normalized in the range [0,1]. For $R = G = B$ (achromatic colors), $T = 5/8$ and $S = 0$ are taken.

Fuzzy ART Off-Line Training on the GPU for Skin Recognition

Adaptive Resonance Theory (ART) systems are comprised of three layers or fields of nodes. Fuzzy ART is an extension of the original ART 1 system that incorporates computations from *fuzzy set theory* into the ART network, and thus making it possible to learn

and recognize both analog and binary input patterns (Carpenter, Grossberg & Rosen, 1991). The first field F_0 represents the input pattern; the upper field F_2 represents the active code or category of the input pattern being selected; the middle layer F_1 receives both bottom-up inputs from F_0 and top-down inputs from F_2. The F_0 activity vector is denoted by $\vec{I} = (I_1, \cdots, I_M)$ where each component I_i is within the [0,1] interval. A useful rule for avoiding proliferation of categories is *complement coding*. If $\vec{a}$ represents the on-response of the pattern, each component of the off-response $\vec{a}^c$ is defined as $a_i^c \equiv 1 - a_i$. Then, the complement coded input comes $\vec{I} = (\vec{a}, \vec{a}^c) \equiv (a_1 \cdots a_M, a_1^c \cdots a_M^c)$ and $|\vec{I}| = M$ for every input pattern.

In order to train the ANN for skin recognition, complement coded TS features can be chosen, so that input patterns are defined as $\vec{I} = (\vec{a}, \vec{a}^c) \equiv (T, S, 1-T, 1-S)$. This way, in a GPU implementation, each feature vector can be stored using a single *texel* in an RGBA texture

Each node of the F_2 field has an associated weight vector or *Long-term Memory* (LTM) trace $w_j = (w_{j1}, ..., w_{jM})$ which subsumes information both from bottom-up and top-down weight vectors. Initially, all weights are set to one, so each category is said to be *uncommitted*. When a category is first selected it becomes *committed* and the corresponding node in F_2 re-adapts its associated weights w_j. For each input $\vec{I}$ and F_2 node j, the *choice function* T_j, is defined by:

$$T_j(\vec{I}) = \frac{|\vec{I} \wedge w_j|}{\alpha + |w_j|},$$

(4)

where the fuzzy MIN operator $\wedge$ is defined by $(p_i \wedge q_i) \equiv \min(p_i, q_i)$ and the norm $|\cdot|$ is defined by $|\vec{p}| \equiv \sum_{i=1}^{M} |p_i|$. The system is said to make a *category choice* when at least one F_2 node becomes active when an input pattern is presented at the F_0 entrance.

The category choice is indexed by J, where $T_J = \max(T_j : j = 1...N)$. Then, w_J is said to be a *fuzzy subset* of $\vec{I}$ and it is fed down from F_2 in order to measure its resemblance to the input pattern $\vec{I}$. The system enters in *resonance* if the *match function* meets the *vigilance criterion*:

$$\frac{|\vec{I} \wedge w_j|}{|\vec{I}|} \geq \rho.$$

(5)

Fuzzy ART implementations on the CPU sequentially compute the activity for every node in field F_2 (4). Then, a sort operation is executed in order to know which neuron is most fired by the input pattern. If the category stored in this neuron resembles enough to the input pattern (5), its associated weights are updated with the new information; otherwise, next most fired neuron must be analyzed. Fuzzy ART implementations following a *stream programming model* can compute the activity of every output neuron simultaneously. Moreover, on a GPU it is possible to take advantage of processing units devised to operate on vector data, and thus to select the most fired neuron whose *match rate* is bigger than a *vigilance parameter ρ* at once. By using complement coding we drastically reduce proliferation of categories and force $|\vec{I}|$ to be constant ($|\vec{I}| = M = 2$) for every input pattern. This also allows for avoiding extra computing when calculating the *match rate* (5). In case vigilance criterion is met and training is switched on, vector w_J must be updated using:

$$w_J^{new} = \beta(\vec{I} \wedge w_J^{old}) + (1 - \beta)w_J^{old}.$$

(6)

In a GPU implementation of a Fuzzy ART ANN devised for skin recognition LTM traces have 4 components and can be stored in a one-dimensional RGBA texture W^T. This texture should be long enough to contain as many categories as could be committed during training process. However, only first N *texels* containing information from committed neurons must participate in the training process when computing T_j. This can be done on the GPU using *scissoring,* which allows rendering a quad of dimensions 1xN which does not cover the whole texture. Scissoring can be used also for updating just those *texels* that should change during training process (6).

Training patterns can be extracted from images containing skin regions. For the experimental results shown in this paper, skin regions were carefully selected from 3056-image Faces96 database (Spacek, 1996). Skin color distribution was estimated as a normal distribution through the *Minimum Covariance Determinants* (MCD) estimator (Rousseeuw & Driessen, 1999) and a total of 671438 input vectors were selected to train the ANN depending on their *mahalanobis* distance to the mean color of the modeled distribution. The ANN was trained fixing α parameter to 0.001 and varying the *vigilance parameter* in different training tests. Table 1

Table 1. Number of committed categories varying ρ

ρ	0.90	0.93	0.95	0.97
N	9	15	33	71

Figure 1. Skin regions belonging to different committed categories varying ρ

a) RGB	*b) ρ = 0.90*	*d) ρ = 0.95*	*e) ρ = 0.97*

shows the number of committed categories depending on the value of ρ.

The larger the value of the vigilance parameter, the larger the number of committed skin categories by the network as level of resemblance between patterns belonging to different categories increases. Figures from 1b) to 1e) show different regions identified as skin categories by the network with ρ increasing from 0.90 to 0.97 respectively.

Fuzzy ART Real-Time Skin Recognition on the GPU

Once the ANN has been trained, computed LTM traces contain all the information that it is needed for skin recognition. Video sequences that have to be processed can be acquired using a conventional USB Webcam and every new frame can be uploaded to the GPU memory and stored in an RGBA texture. Then, a *shader* can be used to convert *(R,G,B,A)* color space pixels into *(T,S,1-T,1-S)* feature vectors, which will be the input for the Fuzzy ART ANN.

During skin recognition, several input patterns can be categorized in a parallel fashion using every fragment processor available on the GPU. *Category choice* occurs through the execution of a *shader* for N times, being N the number of categories in field F_2. In each pass, the *activation* of the j*th* output neuron (5) and the *match rate* (6) are computed for every input pattern and rendered into an RGBA output texture, which will also contain the category index associated to each pattern. This RGBA texture and texture containing feature vectors are used as inputs for the next iteration, using the *ping-pong technique*. If the activation in pass j is bigger than the computed activation in pass $j-1$ and the *match criterion* is satisfied, then the *category index* is updated in the output texture. Finally, a post-processing stage can be used to generate an image where those pixels not belonging to any skin category are not rendered to the screen. Fig. 2 shows a global scheme of the system and the evolution of the skin recognition process through different *shader passes*.

Rendering both the index of the selected category and the *match rate* to the output texture is useful for analyzing results achieved. Different gray levels on channel R represent different skin categories committed during training process; on channel A, a value in the range [0,1] represents the level of resemblance of every pixel in the original image to the selected skin category.

Figure 3 shows two images categorized by the ANN using different ρ values. As ρ increases, both hit rate and false alarm rate decrease. With ρ = 0.90 almost every skin pixel is correctly recognized, but several non-skin pixels (e.g. from the purple glasses) are included in some skin category by the network. These pixels are correctly not recognized as skin with ρ = 0.97.

Table 2 shows the performance of the system for different resolutions running on a dual-core 3.2 GHz Pentium 4 with 1GB RAM, GeForce 7800GTX 256 MB GPU (containing 24 fragment processors) and a generic webcam able to capture up to 90 fps at resolutions of 640x480 pixels. As the value of ρ and resolution increase, frame rate decreases. The number of frames that can be processed by the network strongly depends on the number of input vectors and the number of committed categories every pixel has to be tested to. Best performance is 270 fps, for a resolution of 320x240 pixels and $\rho = 0.90$.

FUTURE TRENDS

Described implementation of the GPU-based skin recognition system in this article was developed using a combined C++ / OpenGL (Shreiner, Woo, Neider & Davis, 2005) / Cg solution (Randima & Kilgard, 2003), and the algorithm had to be translated into graphics terms so that it could be mapped to the GPU (Harris, 2005). However, newer graphics cards from NVIDIA can be programmed using the CUDA *(Compute Unified Device Architecture)* software development kit. Before CUDA was available GPGPU required the use of a graphics API, which presents the wrong abstraction for general-purpose parallel computation, making GPGPU applications difficult to write, debug, and optimize. CUDA enables direct implementation of parallel computations in the C language using an

Figure 2. Global system architecture
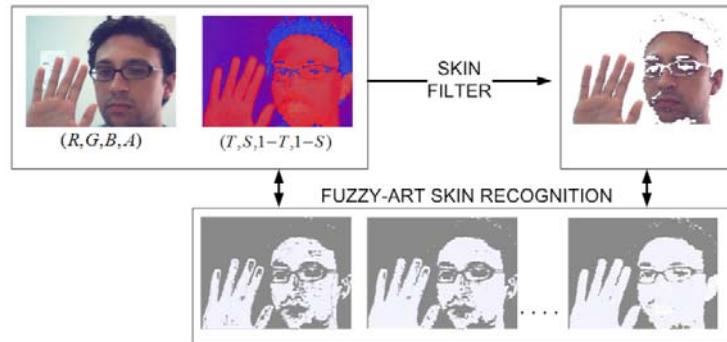
Figure 3. Skin recognition performance varying ρ

| (a) RGB | (b) $\rho = 0.90$ | (c) $\rho = 0.95$ |
| (d) RGB | (e) $\rho = 0.93$ | (f) $\rho = 0.97$ |
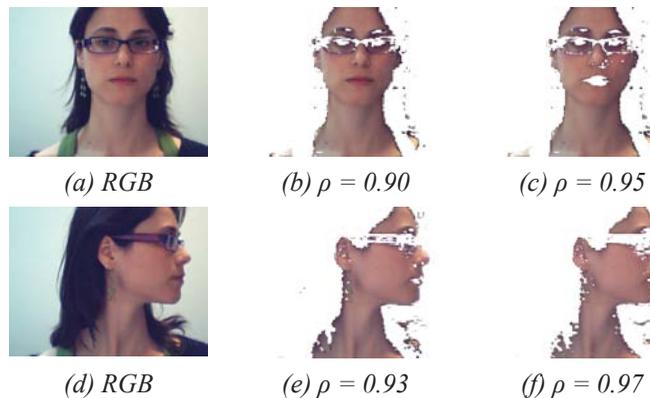
Table 2. Frames per second for different resolutions and ρ values

Resolution	320x240 pixels			352x288 pixels			640x480 pixels		
ρ	0.90	0,95	0,97	0.90	0,95	0,97	0.90	0,95	0,97
fps	270	89	42	212	68	32	71	23	11

API designed for general-purpose computation. It also includes standard FFT and BLAS libraries that will help researchers from different areas to exploit GPUs computational performance.

CONCLUSION

An implementation of a GPU-based Fuzzy ART Neural Network for real time skin recognition was introduced in this paper. This design successfully faces the problem of using a neural network for pattern classification when time is a major requirement. A robust and complete set of skin colors and a good selection of input features (chrominance components of TSL color space) are necessary to train the network so that it can recognize skin in real changing conditions.

Experimental results show system achieves excellent performance with an NVIDIA 7800GTX GPU video card, which includes 24 fragment *shaders* in the pipeline. Fuzzy ART skin recognition on the GPU can be the first stage in a complex computer vision application, like a human-machine interface or a video vigilance system.

REFERENCES

Bernhard, F., & Keriven, R. (2006). Spiking neurons on gpus. In Peter M.A. Sloot Vassil N. Alexandrov, Geert Dick van Albada and Jack Dongarra, editors, *Computational Science – ICCS 2006*, *LNCS* 3994, pp. 236–243. Springer.

Carpenter, G. A., Grossberg, S., & Rosen, D. B. (1991). Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4(6), pp. 759–771.

Duan-sheng, C., & Zheng-kai, L. (2003). A novel approach to detect and correct highlighted face region in color image. In: *AVSS '03: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, Washington, DC, USA, IEEE Computer Society (7).

Fung, J. (2005). Computer vision on the gpu. In Matt Pharr, editor, *GPU Gems 2*, chapter 40, pp 649–665. Addison Wesley.

Hsieh, I.S., Fan, K.C., & Lin, C. (2005). A statistic approach to the detection of human faces in color nature scene. *Pattern Recognition* (35), pp. 1583–1596.

Karlekar, J. & Desai, U.B. (1999). Finding faces in color images using wavelet transform. In *Proceedings. International Conference on Image Analysis and Processing*, pp.1085-1088.

Luo, Z., Liu, H. & Wu, X. (2005). Artificial neural network computation on graphic process unit. In *IJCNN '05: Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, pages 622–626, Montreal, Canada.

Martínez-Zarzuela, M., Díaz, F.J., González, D., Díez, J.F. & Antón, M. (2007). Real Time GPU-based Fuzzy ART Skin Recognition. In Kok, J.N., Koronacki, J., López de Mantaras, R., Matwin, S., Mladenic, D. & Skowron, A., editors: *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Warsaw, Poland. Springer LNCS (4702), pp. 548–555.

Oh, K-S. & Jung, K. (2004). GPU implementation of neural networks. *Pattern Recognition*, 37(6), pp. 1311–1314.

Owens, J. (2005). Streaming architectures and tecnology trends. In Matt Pharr, editor, *GPU Gems 2*, chapter 29, pp. 457–470. Addison Wesley.

Phung, S.L., Bouzerdoum, A., & Chai, D. (2005). Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(1), pp. 148–154.

Randima, F., & Kilgard, M. (2003). *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*. Addison Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Rousseeuw, P.J. & Driessen, K.V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41(3), pp. 212–223.

Sahbi, H. & Boujemaa, N. (2000). From coarse to fine skin and face detection. In *Proceedings of the Eighth ACM international Conference on Multimedia*, pp. 432–434, California, United States.

Spacek, L. (1996). Faces96 db, http://cswww.essex.ac.uk/mv/allfaces/faces96.html, accessed: October 2007.

Shreiner, D., Woo, M., Neider, J., & Davis, T. (2005). Opengl Programming Guide: the Official Guide to Learning Opengl, Version 2 (5th Edition). Addison-Wesley Professional.

Steinkraus, D., Simard, P.Y., & Buck, I. (2005). Using gpus for machine learning algorithms. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pp. 1115–1119, Washington, DC, USA. IEEE Computer Society.

Terrillon, J., David, M., & Akamatsu, S. (1998). Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments. *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*, pp. 112–117, Nara, Japan.

KEY TERMS

ART *(Adaptive Resonance Theory)***:** Learning theory developed by S. Grossberg that is used in competitive neural systems and includes short-term-memory (STM) and long-term-memory (LTM) processes.

CUDA: A GPGPU technology that allows a programmer to use the C programming language to code algorithms for execution on the GPU. CUDA requires an NVIDIA GPU and special stream processing drivers.

Fuzzy ART: Evolution of the ART1 neural network capable of learning normalized analog input patterns in an unsupervised way through the use of fuzzy operators.

GPGPU *(General-Purpose Computation on GPUs)***:** A recent trend in computer science consisting in the use of the *Graphics Processing Unit* (GPU), for doing expensive computational tasks rather than just computer graphics.

GPU *(Graphics Processing Unit)***:** A dedicated graphics rendering device very efficient at manipulating and displaying computer graphics, thanks to its highly parallel structure.

Heterogeneous Multi-Core Computing: Design and analysis of algorithms and applications for heterogeneous multi-core processor architectures (e.g. IBM Cell processor).

Homogeneous Multi-Core Computing: Design and analysis of algorithms and applications for homogeneous multi-core processor architectures (e.g. GPUs).

Moore's Gap: Refers to the relatively modest incremental performance gains brought about by the increased number of transistors on current uni-processor dies despite increases in clock speeds.

Stream Processing: A paradigm for the execution of parallel processing operations exploiting *data-level parallelism* rather than *task-level parallelism* that provides incredible performance with minimal programming effort.

TSL Color Space: Color space based on *Intensity Hue Saturation* (IHS) color model. A color in this space is specified by *Tint* (T), *Saturation* (S) and *Luminance* (L) values.

A Study of the Performance Effect of Genetic Operators

Pi-Sheng Deng
California State University at Stanislaus, USA

INTRODUCTION

Performance of genetic algorithms (GAs) is mainly determined by several factors. Not only the genetic operators affect the performance of a GA with varying degrees, but also the parameter settings for genetic operators interact in a complicated manner with each other in influencing a GA's performance. Though many studies have been conducted for this cause, they failed to converge to consistent conclusions regarding the importance of different genetic operators and their parameter settings on the performance of GAs. Actually, optimizing the combinations of different strategies and parameters for different problem types is an *NP*-complete problem in itself, and is still an open research problem for GAs (Mitchell, 1996).

Recognizing the intrinsic difficulties in finding universally optimal parameter configurations for different classes of problems, we advocate the experience-based approach to discovering generalized guiding rules for different problem domains. To this end, it is necessary for us to gain a better understanding about how different genetic operators and their parameter combinations affect a GA's behavior. In this research, we systematically investigate, through a series of experiments, the effect of GA operators and the interaction among GA operators on the performance of the GA-based batch selection system as proposed in Deng (2007). This paper intends to serve as an initial inquiry into the research of useful design guidelines for configuring GA-based systems.

PARAMETER CONFIGURATION FOR GENETIC OPERATORS

It is commonly believed that crossover is the major operator of GAs, with mutation preventing the population from early convergence to a certain solution before an extensive exploration of other candidate solutions

is made (Holland, 1992a). Crossover enables GAs to focus on the most promising regions in a solution space; however, mutation alone does not advance the search for a solution. Crossover is also a more robust constructor of new candidate solutions than mutation (Spears, 1993).

However, Muhlenbein (1992) argues that the power of mutation has been underestimated in traditional GAs. According to Mitchell (1996), it is not a choice between crossover or mutation but rather the balance among crossover, mutation, and other factors, such as selection, that is all important. The correct balance also depends upon the details of the fitness function and the encoding. Furthermore, crossover and mutation vary in relative usefulness over the course of a run. Actually, the theretical analysis of crossover is still to a large extent an open problem (Back, *et al.*, 1997).

In addition to the GA operators, the population size also affects the performance of GAs. The specification of the population size affects the diversification of the population body and the implicit parallelism of a GA, and will thus affect the quality of the generated solutions and the performance of the solution-generating process. Choosing an appropriate population size for a GA is a necessary but difficult task for GA users. Usually, the parameter settings for most GA applications are based on De Jong's recommendations (De Jong, 1975). According to De Jong's experiments with five problems in function minimization, the best population size was 50~100, the best crossover rate was about 0.6, and the best mutation rate was 0.001. In a later study, Spears & De Jong (1991) suggested a wider range for the crossover rate as 0.5~0.8. Mitchell (1996) also observed that it was common in GA applications to set crossover rate at 0.7~0.8.

However, Schaffer *et al*. (1991) asserted that the best settings for population size, crossover rate, and mutation rate were independent of the problems. In their study of a small set of numerical optimization problems, a very small population of size 20~30 with

a large crossover rate ranging from 0.75 to 0.95, and with a very small mutation rate ranging from 0.005 to 0.01 would produce the best perofrmance. Grefenstette (1993) also reached similar conclusions in his study of parameter optimization for GAs, and suggetsed the following settings: population size 30, crossover rate 0.95, and mutation rate 0.01. While Schaffer *et al*. (1991) and Grefenstette (1993) advocated a very small population size, Goldberg (1989) and Liao & Sun (2001) argued for a much larger population size.

From the above discussion, the diverse recommendations on population size seem to indicate that population size interacts with some other factors not included in the previous research. In this paper, we investigated the effect of interaction among different parameters on a GA's performance. However, the choice of mutation rate needs to take into account, at least, the task complexity of an application. According to Mitchell (1996), it is impossible to specify an optimal setting for parameters in all different applications.

EXPERIMENTAL DESIGN

We focus mainly on investigating the effects of different combinations of parameter settings for genetic operators on our GA's performance, and compare our results with the claims made by previous research. We discuss the factors and parameter settings experimented in this study as below:

- **Task complexity:** Since the length of the solution string is usually a function of the complexity of the problem, we experiment with different parameter settings for two batch selection tasks of different complexity levels. One task has 30 products to be manufactured, 10 available tools, and 8 available machines. The other less complex task has 12 products, 6 available tools, and 4 available machines.

- **Representation scheme:** In this paper, we consider a common situation in FMSs in which if a product is selected in a batch for manufacturing, the entire quantity specified in the production table must be produced in the shift. Under this assumption, our batch selection task becomes a pseudo-Boolean optimization problem. This enables us to use a single binary bit to represent a component in a candidate solution. Therefore,

each candidate solution to the batch selection task can be encoded as a binary string of fixed length P, where P is the cardinality of the entire set of products under consideration.

- **Population size:** If the population size is too small, the GA will converge too quickly to find the optimal solution; however, if the population size is too large, the computation cost will be prohibitive. In this research, we investigate the effect of population sizes 10, 100, and 200, representing Small, Medium, and Large, on generating solutions for our batch selection problem.

- **Selection strategy:** We adopted the elitism strategy so that the best candidate solutions at each generation could be retained for the next generation. Though elitism is used to prevent the elite solution strings in a population from being altered by crossover or mutation, retaining too many elite individuals might cause the domination of the entire population by suboptimal, though highly fit, solution strings. This might lead to degeneration for the population eventually. The usual practice is to retain a small number of elite candidate solutions (Goldberg, 1989). In this research our system preserves two fittest candidate solutions on each iteration of forming new population.

- **Crossover parameter:** We adopt the standard crossover operator, i.e., the one-point crossover. The crossover rate is the probability that the crossover operator will be applied to a pair of candidate solutions selected for reproduction. In order to re-examine the different claims by previous research on the importance of different crossover rates, we experiemnt with three different crossover rates: 0.1, 0.5, and 0.9, representing three levels: High, Medium, and Low.

- **Mutation parameter:** The parameter *mutation rate* is used to control the rate of diversification via probabilistic conversion of each bit value in a candidate solution. However, a mutation rate approaching 1 will theoretically lead to a completely stochastic search with no succession from generation to generation. The usual practice is applying an occasional mutation to make a random change in the elements of a solution string. There are also various conclusions from previous research regarding the mutation rate. In this research we experiment with three different mutation rates: 0.001, 0.01, and 0.5, representing three levels: Low, Medium, and High.

- • **Termination criteria:** A termination criterion can be a specified maximum number of generations, a target objective function value, a convergence threshold, or a lack of improvement in the best solution over a specified number of generations. In this research, our system will terminate when there is no improvement in the best solutions over 50 consecutive generations.

PERFORMANCE ANALYSIS FOR COMPUTATIONAL EXPERIMENT

The optimal-batch search process is conducted for each parameter combination until 50 feasible solutions are generated. We experimented with the combinations of three population sizes, three levels of the crossover rate, and three levels of the mutation rate for two tasks. Altogether, we conducted 4381 times of experimentation in generating 2700 feasible solutions. Performance analysis for each parameter setting is discussed for each of the two tasks under study.

Performance Analysis for the Task with Higher Complexity

The result of our experiment for the higher-complexity task is shown in Table 1. As suggested by Mitchell (1996), a GA's behavior had better be understood and described by macroscopic statistics, such as mean fitness in the population. Therefore, we compute the average performance and standard deviation in Table 1. The average performance of each parameter combination is obtained by averaging the best results over 50 feasible solutions.

From Table 1, we find out that for all different combinations of crossover rates and mutation rates, the average performance for *Pop Size* = 200 is always the best. This implies that there is no strong interaction among the three parameters. In addition, the standard deviation column indicates that when the population size is larger, the fluctuation of performance from different runs of generating optimal solutions tends to be smaller. Though populations of size 200 would yield the best performance in our experiment, the number of runs of simulation for generating 50 feasible solutions is also the largest. Populations of size 10 are most likely to generate feasible solutions which tend to have the lowest performance.

If we look across all different population sizes, it seems that when the crossover rate is set at a low value, e.g., 0.1, the mutation rate should be set at a very small value for the best result. When the crossover rate is set at a medium or high level, the mutation rate 0.01 favors the performance the most. Overall, there seems to have a tendency that across all levels of population size, *Mutation Rate* = 0.01 and a medium- or high-level crossover rate will generate the best result. This implies that there might have an interaction between the crossover rate and the mutation rate.

Across all different levels of the crossover rate, the combination of *Mutation Rate* = 0.01 and *Pop Size* = 200 tends to consistently yield the best result. This implies that there is a strong interaction between the population size and the mutation rate. Across all levels of the mutation rate, there is also a consistent pattern of effects on the system performance among different combinations of population sizes and crossover rates: the population size 200 combines with high crossover rates in generating the best result in Table 1. This seems

Table 1. Performance of different population sizes under different mutation rates and crossover rates

Crossover Rate	Pop. Size	Mutation Rate = 0.001			Mutation Rate = 0.01			Mutation Rate = 0.5		
		Ave (%)	Stdev (%)	#F/#T	Ave (%)	Stdev (%)	#F/#T	Ave (%)	Stdev (%)	#F/#T
0.1	10	88.25	3.37	50/50	87.60	2.69	50/52	85.54	2.57	50/53
	100	93.02	1.48	50/56	93.25	1.69	50/74	91.41	2.13	50/79
	200	93.76	1.45	50/63	93.70	1.30	50/98	92.65	2.10	50/93
0.5	10	89.43	2.69	50/50	89.64	3.05	50/51	85.86	2.75	50/53
	100	93.21	1.58	50/51	93.37	1.49	50/54	91.68	2.03	50/89
	200	93.79	1.21	50/52	93.86	1.21	50/60	92.19	1.74	50/115
0.9	10	88.73	3.27	50/50	89.78	2.40	50/50	85.63	3.45	50/52
	100	93.39	1.32	50/50	93.58	1.64	50/50	91.48	1.94	50/64
	200	93.81	1.27	50/50	94.30	1.39	50/53	92.40	1.59	50/89

(Note: # F/#T: the ratio of the number of feasible solutions generated to the total number of simulation runs.)

Table 2. Performance of different population sizes under different mutation rates and crossover rates

Crossover Rate	Pop. Size	Mutation Rate = 0.001			Mutation Rate = 0.01			Mutation Rate = 0.5		
		Ave (%)	Stdev (%)	#F/#T	Ave (%)	Stdev (%)	#F/#T	Ave (%)	Stdev (%)	#F/#T
0.1	10	88.24	3.13	50/50	89.30	2.51	50/51	90.25	1.60	50/50
	100	91.25	0.68	50/57	91.15	0.81	50/57	91.87	0.25	50/124
	200	91.32	0.56	50/55	91.39	0.62	50/69	91.89	0.21	50/309
0.5	10	89.37	2.36	50/50	89.65	1.99	50/50	89.90	1.58	50/51
	100	91.22	0.60	50/50	91.22	0.60	50/50	91.74	0.41	50/108
	200	91.29	0.66	50/50	91.64	0.48	50/50	91.91	0.15	50/431
0.9	10	89.00	2.48	50/50	89.55	1.87	50/50	90.17	1.64	50/51
	100	91.17	0.68	50/50	91.26	0.71	50/50	91.76	0.40	50/99
	200	91.45	0.56	50/50	91.49	0.53	50/50	91.89	0.21	50/518

(Note: # F/#T: the ratio of the number of feasible solutions generated to the total number of simulation runs.)

to indicate there is a strong interaction between the population size and the crossover rate.

Performance Analysis for the Task with Lower Complexity

The result of our experiment for the other task is shown in Table 2. From Table 2, *Mutation Rate* 0.5 tends to produce the best result with the smallest deviation. The same observations also hold for the *Pop Size* = 200. This indicates that there is <u>no</u> strong interaction among these three parameters. However, *Mutation Rate* = 0.5 and *Pop Size* = 200 also have the lowest number of feasible solutions. On the other hand, *Pop Size* 10 and *Mutation Rate* 0.001 are most likely to generate feasible solutions which tend to have the lowest performance. From Table 2, we cannot identify any consistent pattern of performance for the crossover rate. Similar to the previous case, the interaction between the crossover rate and the mutation rate does not have a consistent pattern of influence on the system performance across different population sizes. This implies the lack of strong interaction between the crossover rate and the mutation rate for the current case.

Overall, the combination of *Pop Size* = 200 and *Mutation Rate* = 0.5 seems to give the best result for all different levels of the crossover rate. This implies that there is a <u>significant</u> interaction between the population size and the mutation rate. However, we cannot identify a consistent pattern for the combination of the population size and the crossover rate or the mutation rate and the crossover rate. This implies that there is <u>lack</u> of an interaction within these two pairs of parameters.

FUTURE TRENDS AND CONCLUSION

Though Schaffer *et al*. (1991) and Grefenstette (1993) advocate a very small population size, our analyses for both tasks of high complexity and low complexity indicate that larger populations will generally favor the performance of our batch selection system more than smaller populations. Our result is consistent with Liao & Sun (2001). With the availability of a larger pool of diverse schemata in a larger population, our GA system will have a broader view of the "landscape" (Holland, 1992b) of the solution space, and is thus more likely to contain representative solutions from a large number of hyperplanes. This advantage gives a GA more chances of discovering better solutions in the solution space. However, Davis (1991) argues that the most effective population size is dependent upon the nature of the problem, the representation formalism, and the GA operators. We plan to analyze the GA performance for another application domain so that we can be more conclusive on the issue of the effective population size.

Though the solution performance of small populations is lower than that of large populations, the efficiency of small populations in generating feasible solutions, i.e., the ratio of number of feasible solutions to the total number of runs required to generate a certain number of feasible solutions, is indeed better than large populations, especially when the mutation rate is high. This can be evidenced by the #F/#T columns of Tables 1 and 2. In this sense, Schaffer *et al*. (1991) and Grefenstette (1993) are correct in their recommendation. This might be due to the fact that small populations have higher probability of developing the premature convergence problem.

Our analysis shows that our two tasks do not agree on the recommendation for the mutation rate. The task with higher complexity prefers a very small mutation rate, especially 0.01; while the less complicated task prefers a very large mutation rate, such as 0.5. In addition, high crossover rates will be better for complex tasks; while there is no conclusion for simple tasks. Contrary to the general belief regarding the major role of crossover, *we did not find out crossover was as a determinant factor as population size or mutation rate in influencing the system performance*. Part of our findings is similar to that of Pendharkar & Rodger (2004), who compared the performance of different types of crossover operators, including arithmetic, uniform, and one-point operators, for the design of GA-based artificial neural networks and found no significant difference among them. In addition, our findings on the role of mutation rate for tasks of different complexity complement Muhlenbein (1992) who contends that the power of mutation has been underestimated in traditional GAs.

Our analysis also shows mutation and crossover interact with the population size in different ways. The effect of mutation is strongly influenced by the population size in both tasks. For the task with higher complexity, the combination of a very large population size, such as 200, and a small mutation rate, such as 0.01, tends to generate a very good result. However, the less complex task needs a very large population and a very large mutation rate, such as 0.5, in order to yield the best results. On the other hand, the interaction between crossover and the population size is only found with the task of high complexity, and the interaction between mutation and crossover is barely found with the task of higher complexity only. More research work needs to be performed in order to understand better how the effetcs of crossover and mutation depend upon other details of a GA, such as the population size, the application domain, the fitness function, encoding, and selection.

REFERENCES

Back, T., Hammel, U., & Schwefel, H. (1997). Evolutionary Computation: Comments on the History and Current State. *IEEE Transactions on Evolutionary Computation*. (1)1, 3-17.

Davis, L. (Editor) (1991). *Handbook of Genetic Algorithms*. New York, NY: Van Nostrand Reinhold.

De Jong, K.A. (1975). *An Analysis of the Behavior of A Class of Genetic Adaptive Systems*. Ph.D. thesis, University of Michigan, Ann Arbor, MI.

Deng, P-S. (2009). Applying Genetic Algorithms to Optimization Modeling. In Dopico J.R.R, de la Calle, J. D. & Sierra, A.P. (Eds.), *Encyclopedia of Artificial Intelligence*, Hershey, PA: IDEA.

Goldberg, D.E. (1989). *Genetic Algorithm in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.

Grefenstette, J.J. (1993). Introduction to the Special Track on Genetic Algorithms. *IEEE Expert*. October, 5-8.

Holland, J. (1992a). A*daptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press.

Holland, J. (1992b). Genetic Algorithms. *Scientific American*. July, 66-72.

Liao, Y.H., & Sun, C.T. (2001). An Educational Genetic Algorithms Learning Tool. *IEEE Transactions on Education*. (44)2, 415-423.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

Muhlenbein, H. (1992). How Genetic Algorithms Really Work: Mutation and Hillclimbing. *Parallel Problem Solving From Nature* 2, Manner, R., & Manderick, B. (Editors), North-Holland.

Pendharkar, P.C., & Rodger, J.A. (2004). An Empirical Study of Impact of Crossover Operators on the Performance of Non-binary Genetic Algorithm Based Neural Approaches for Classification. *Computers & Operations Research*. (31) 481-498.

Schaffer, J.D., Caruana, R.A., Eshelman, L.J., & Das, R. (1991). A Study of Control Parameters Affecting Online performance of Genetic Algorithms for Function Optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, Schaffer, J.D. (Editor), San Mateo, CA: Morgan Kaufmann.

Spears, W.M., & De Jong, K.A. (1991). On the Virtues of Parameterized Uniform Crossover. *Proceedings of the Fourth International Conference on Genetic Algorithms*, Belew, R.K., & Booker, L.B. (Editors), San Mateo, CA: Morgan Kaufmann.

Spears, W.M. (1993). Crossover or Mutation? *Foundations of Genetic Algorithms* 2, Whitley, L.D. (Editor), San Mateo, CA: Morgan Kaufmann.

KEY TERMS

Batch Selection: Selecting the optimal set of products to produce, with each product requiring a set of resources, under the system capacity constraints.

Fitness Function: The objective function of the GA for evaluating a population of solutions.

Genetic Operators: Selection, crossover, and mutation, for combining and refining solutions in a population.

Implicit Parallelism: A property of the GA which allows a schema to be matched by multiple candidate solutions simultaneously without even trying.

Landscape: A function plot showing the state as the "location" and the objective function value as the "elevation".

NP-Complete Problems: The hardest problems in the class NP—the class of nondeterministic polynomial problems.

Schemata: A general pattern of bit strings that is made up of 1, 0, and #, used as a building block for solutions of the GA.

Supervised Learning of Fuzzy Logic Systems

M. Mohammadian
University of Canberra, Australia

INTRODUCTION

Conventionally modelling and simulation of complex nonlinear systems has been to construct a mathematical model and examine the system's evolution or its control. This kind of approach can fail for many of the very large non-linear and complex systems being currently studied. With the invention of new advanced high-speed computers and the application of artificial intelligence paradigms new techniques have become available. Particularly neural networks and fuzzy logic for nonlinear modelling and genetic algorithms [Goldberg, D. (1989)] and evolutionary algorithms for optimisation methods have created new opportunities to solve complex systems [Bai, Y., Zhuang H. and Wang, D. (2006)].

This paper considers issues in design of multi-layer and hierarchical fuzzy logic systems. It proposes a decomposition technique for complex systems into hierarchical and multi-layered fuzzy logic sub-systems. The learning of fuzzy rules and internal parameters in a supervised manner is performed using genetic algorithms. The decomposition of complex nonlinear systems into hierarchical and multi-layered fuzzy logic sub-systems reduces greatly the number of fuzzy rules to be defined and improves the learning speed for such systems. In this paper a method for combining subsystems to create a hierarchical and multilayer fuzzy logic system is also described. Application areas considered are - the prediction of interest rate, unemployment rate predication and electricity usage prediction.

Genetic Algorithms can be used as a tool for design and generation of fuzzy rules for a fuzzy logic system. This automatic design and generation of fuzzy rules, via genetic algorithms, can be categorised into two learning techniques namely, supervised and unsupervised. In supervised learning there are two distinct phases to the operation. In the first phase each individual is assessed based on the input signal that is propagated through the system producing output respond. The ac-

tual respond produced is then compared with a desired response, generating error signals that are then used as the fitness for the individual in the population of genetic algorithms. Supervised learning has successfully applied to solve some difficult problems. In this paper design and development of a genetic algorithm based supervised learning for fuzzy models with application to several problems is considered. A hybrid integrated architecture incorporating fuzzy logic and genetic algorithm can generate fuzzy rules that can be used in a fuzzy logic system for modelling, control and prediction.

Fuzzy logic systems typically have a knowledge base consisting of a set of rules of the form

If (x_1 is $A_1^{'}$ and x_2 is $A_2^{'}$ and … and x_n is $A_n^{'}$)
Then (z_1 is $B_1^{'l}$ else z_2 is $B_2^{'l}$ else … else z_m is $B_m^{'l}$)

where $A_k^{'l}$; $k = 1, …, n$ are normalised fuzzy sets for n input variables x_k, $k = 1$;..., n, and where $B_k^{'l}$, k ; $k = 1, …, m$ are normalised fuzzy sets for m output variables z_k, $k = 1, …, m$. The heart of the fuzzy logic system is the inference engine that applies principles of intelligent human reasoning to interpret the rules to output an action from inputs. There are many types of inference engines in the literature, including the popular Mamdani inference engine, [Bai, Y., Zhuang H. and Wang, D. (2006)].

Given a fuzzy rule base with M rules and n antecedent variables, a fuzzy controller as given in Equation 1 uses a singleton fuzzifier, Mamdani product inference engine and centre average defuzzifier to determine output variables, has the general form for a single output variable, say z_1

$$z_1 = \frac{\sum_{l=1}^{M} y_k^l \left(\prod_{i=1}^{n} \mu A_l^i (x_i) \right)}{\sum_{l=1}^{M} y_k^l \left(\prod_{i=1}^{n} \mu A_l^i (x_i) \right)}$$

(1)

where y_k^l are centres of the output sets B_k^l and membership function μ defines for each fuzzy set A_i^l the value of x_i in the fuzzy set, namely, $\mu\, A_1^i(x_i)$). Common shapes of the membership function are typically, triangular, trapezoidal and Gaussian. A first step in the construction of a fuzzy logic system is to determine which variables are fundamentally important. It is known that the total number of rules in a system is an exponential function of the number of system variables [Raju G. V. S. and Zhou, J. (1993), Kingham, M., Mohammadian, M, and Stonier, R. J. (1998)]. In order to design a fuzzy system with the required accuracy, the number of rules increases exponentially with the number of input variables and their associated fuzzy sets to the fuzzy system. A way to avoid the explosion of fuzzy rule bases in fuzzy logic systems is to consider Hierarchical Fuzzy Logic systems [Raju G. V. S. and Zhou, J. (1993)]. Hierarchical fuzzy logic systems have the property that the number of rules needed to construct the fuzzy system increases only linearly with the number of variables in the system.

The idea of hierarchical fuzzy logic systems is to put the input variables into a collection of low-dimensional fuzzy logic systems, instead of creating a single high dimensional rule base for a fuzzy logic system. Each low-dimensional fuzzy logic system constitutes a level in the hierarchical fuzzy logic system. Assume that there are n input variables $x_1,...,x_n$ then the hierarchical fuzzy logic system is constructed as follows [Raju G. V. S. and Zhou, J. (1993)]

- The first level fuzzy rule base for fuzzy system with n_1 input variables $x_1,...,x_n$ which is constructed from the rules

  If x_1 is A_1^t and ... and x_n is A_{n1}^t, Then y_1 is B_1^l

  where $2 \leq n_1 \leq n$, and $l = 1,2,..., M_1$.
- The i'th level ($i > 1$) fuzzy rule base for a fuzzy system with $n_i + 1$ ($n_i \geq 1$) input variables, which is constructed from the rules

  If $x_{N_{i+1}}$ is $A_{N_{i}\ 1}$ and ... and $A_{N_{i}\ n_i}^l$ and y_{i-1} is   Then y_i is B_i^l

  where

  $$N_i = \sum_{j=1}^{i-1} N_j \,,$$

and $l = 1,2, ... , M_i$
- The construction of fuzzy rule bases for fuzzy systems continues until $i=l$ such that

  $$N_i = \sum_{j=1}^{i-1} N_j = n \,,$$

  that is, until all the input variables are used in one of the levels.

The first level has n_1 input variables $x_1,...,x_n$ with one output variable y_1, which is then sent to the second level as input. In the second level another n_2 variables $x_{n_{i+1}},...,x_{n_1+n_2}$ and the variable y_1 are combined to produce the output variable y_2, which is then sent to the third level. This procedure continues until all the variables $x_1,...,x_n$ are used [Raju G. V. S. and Zhou, J. (1993), Kingham, M., Mohammadian, M, and Stonier, R. J. (1998), Magdalena, L. (1998), Cordon, O., Herrera, F., Hoffmann, F. and Magdalena, L. (2001)]. The number of rules in a hierarchical fuzzy logic system is a linear function of the number of input variable and their associate fuzzy sets [Kingham, M., Mohammadian, M, and Stonier, R. J. (1998)]. Other ways to reduce the fuzzy rules of a fuzzy logic system are

1. Fusing variables before input into the inference engine, thereby reducing the number of rules in the knowledge base,
2. Grouping the rules into prioritised levels to design hierarchical or multi-layered structures,
3. Reducing the size of the inference engine directly using notions of passive decomposition of fuzzy relations,
4. Decomposing the system into a finite number of reduced-order subsystems, eliminating the need for a large-sized inference engine.
5. Reducing the number of fuzzy sets of each input variable, thereby reducing the number of rules in the knowledge base of fuzzy logic system.

Using hierarchical fuzzy logic systems the typically the most influential parameters are chosen as the system variables in the first level, the next most important parameters are chosen as the system variables in the second level, and so on, [Raju G. V. S. and Zhou, J. (1993)]. In this hierarchy, the first level gives an approximate output which is then modified by the second level rule set, this procedure can be repeated

in succeeding levels of hierarchy. The number of rules in a complete rule set is so reduced to a linear function of the number of variables, but this number may still be high. Further, given that different hierarchical and multi-layered structures can exist, how can the fuzzy knowledge base and associated parameters in each layer be effectively learnt? A learning approach based on genetic algorithms is discussed in this paper for the determination of these knowledge bases and associated parameters.

VARIABLE SELECTION, RULE BASE LEARNING AND DECOMPOSITION

Interest Rate Prediction

In [Kingham, M., Mohammadian, M, and Stonier, R. J. (1998)], the authors used hierarchical fuzzy logic structures and multi-layered neural network structures for modelling and prediction of the Australian interest rate with 14 input variables, on actual data of key economic indicators that was a limited data set. Using expert knowledge from an economist the following input variables were chosen and placed into 5 different groupings, namely,

1. Employment (Job Vacancies, Unemployment Rate)
2. Country (Gross Domestic Product, Consumer Price Index)
3. Savings (Household Saving Ratio, Home Loans, Average Weekly Earnings)

4. Foreign (Current Account, RBA Index, Trade Weighted Index)
5. Company (All Industrial Index, Company Profit, New Motor Vehicles)

which then were formed into a two layered fuzzy system, see Figure 1.

The current interest rate was input into each of the five fuzzy systems in the first layer and the final output of the second layer was the predicted interest rate. It is assumed that the first layer gives a first iteration of the new interest rate and they are input into the second layer. But the output variables from the first layer do not necessarily have to be identified with the interest rate. Assuming there are five membership sets for all variables, including those entering the second layer, there are 5250 fuzzy rules in this structure. If all fourteen variables were input into a single layer fuzzy logic system structure there would be some 6 million rules (5^{16}). Hence there is a considerable reduction in the number of rules for this simple two layered hierarchical fuzzy logic system structure. But it is clear that this in not the only decomposition that could have been formed in grouping the variables, or in number of levels of the multi-layered structure. A genetic algorithm was used to learn the rules in this fuzzy system, and it was found that the hierarchical fuzzy logic system structure was accurate [Kingham, M., Mohammadian, M, and Stonier, R. J. (1998)]. Further research on this problem discussing different hierarchical fuzzy structures of three, four and five layers, and the learning of the fuzzy rule bases, was considered and can be found in [Mohammadian, M. and Kingham, M. (2004)].

Figure 1. Interest rate prediction

1512
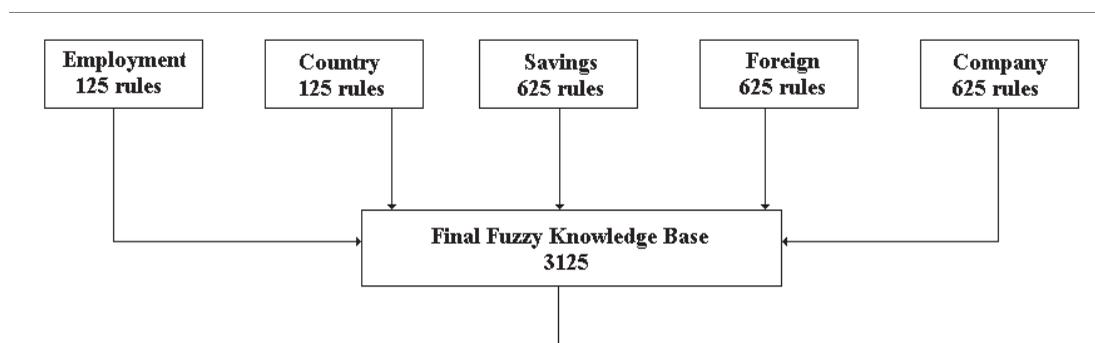
However there is still a question, Does a two layer hierarchical fuzzy logic system structure provides the best solution? To answer this question, one can start building three, four layer hierarchical fuzzy logic system structure by trial and error to possibly find the correct number of layers required. This could be cumbersome problem [Mohammadian, M. and Kingham, M. (2004)]. Genetic algorithms can be used to solve this problem by determining the number of layer in the hierarchical fuzzy logic system and the correct combination of fuzzy knowledge bases for each layer.

A genetic algorithm is developed in such a way to provide the possible best architecture for designing hierarchical fuzzy logic systems for prediction of interest rate in Australia [Mohammadian, M. (2002)]. Using the economic indicators five fuzzy logic systems were developed as described above. Genetic algorithms were

then used to design and develop a hierarchical fuzzy logic system. The hierarchical fuzzy logic system developed was then used to predict interest rate. For each of these group (as described earlier), the current quarter's interest rate is included in the indicators used.

For encoding and decoding of the hierarchical fuzzy logic system, first a number is allocated to each fuzzy logic system developed from group of indicators. For this simulation the number allocated to each group is shown below

1 = Employment, 2 = Country, 3 = Savings, 4 = Foreign, 5 = Company

The number of layers and the fuzzy logic system/s for each layer is determined by genetic algorithms. Genetic algorithms randomly encode each fuzzy logic

Figure 2. A three-layer hierarchical fuzzy logic system – 3125 fuzzy rules
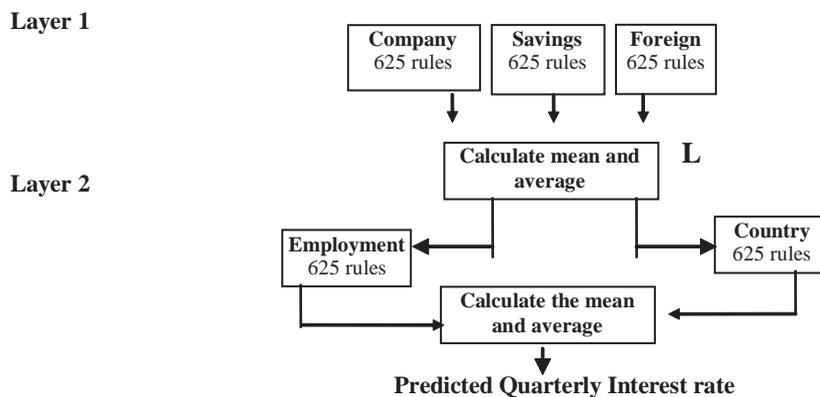
Figure 3.
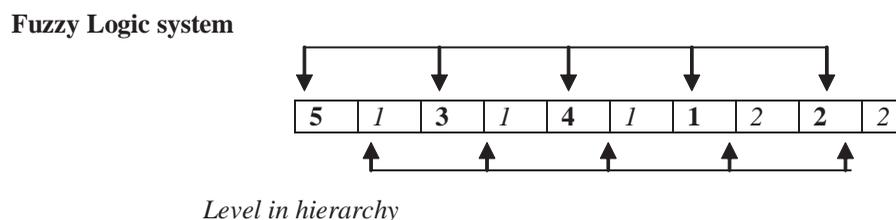
system into a number ranging from 1 to 5 for all possible combinations of the fuzzy logic systems. The level in the hierarchy in which a fuzzy logic system is allocated to, is also encoded in each string representing an individual in a population of genetic algorithms. A string is encoded this way can be represented as Figure 3.

Each individual string is then decoded into a hierarchical fuzzy logic system that defines the fuzzy logic system/s for each level of the hierarchical fuzzy logic system. The above string once decoded will provide a hierarchical fuzzy logic system as shown in Figure 2 above. The set of hierarchical fuzzy logic systems thus developed, are evaluated and a fitness value is given to each string. We define a satisfactory hierarchical fuzzy logic system as one whose fitness value (predicated interest rate) differs from the desired output of the system (in this case the actual interest rate) by a very small value. A calculated the average error of the system was used for the training set and tests sets using the following formula [Mohammadian, M. and Stonier, R. J. (1998)]

$$E = \frac{\sum_{i=1}^{n} abs(Pi - Ai)}{n}$$

where E is the average error, *Pi* is the Predicted interest rate at time period *i*, *Ai* is the actual interest rate for the quarter and *n* is the number of quarters predicted. By using genetic algorithms to design and develop hierarchical fuzzy logic system good results were obtained. The hierarchical fuzzy logic systems developed using genetic algorithms predict the interest rate to different degree of accuracy. It is however interesting to see that genetic algorithms is capable of providing different hierarchical fuzzy logic system structures for predicting the interest rate. It should be noted that genetic algorithm is also capable of finding the number of layers in hierarchical fuzzy logic system.

Prediction of Unemployment Rate

In [Mohammadian, M., Nainar, I. and Kingham, M. (1997)] a fuzzy logic system was developed for the supervised learning in predicting quarterly Unemployment rate in Australia. The following economic indicators where used as input to the Fuzzy Logic system.

- *The Unemployment Rate is the percentage of the labour force actively looking for work in the country.*
- *Interest Rate which is the indicator we are aiming to predict. The Interest Rate used here is the Australian Commonwealth government 10-year treasury bonds.*
- *Job Vacancies is where a position is available for immediate filling or for which recruitment action has been taken.*
- *Household Saving Ratio is the ratio of household income saved to households disposable income.*

Each input was split into five fuzzy sets giving a total of 625 rules. These rules form the fuzzy knowledge base of the system. A supervised learning strategy using of genetic algorithms [Mohammadian, M., Nainar, I. and Kingham, M. (1997)] was used to find the fuzzy knowledge base for the system. Using simulations it was shown that the fuzzy logic system is able to predict with a great deal of success the quarterly unemployment rate. The results achieved proved that the supervised learning strategy used accurately predicted fluctuations in the unemployment rate, and any small errors in the prediction could be reduced by increasing the training data and allowing the learning algorithm to run longer.

Electricity Load Prediction

In [Mohammadian, M. and Jentzsch, R. (2005)] a hierarchical fuzzy logic system using genetic algorithms for the prediction and modelling of daily electricity load fluctuations. The system is further trained to model and predict electricity consumption for daily peak. There are a number of possible indicators that could be used to predict the electricity load. These indicators that were used in this hierarchical fuzzy logic system are

Electricity load (is the past electricity consumption (hourly)),
Predicted Minimum Temperature is the predicted minimum temperature,
Predicted Maximum Temperature is the predicted maximum temperature,
Actual Minimum Temperature is the actual predicted minimum temperature,

Actual Maximum Temperature is the actual predicted maximum temperature,

Season is one of the four seasons in the year,

Day of the week is one of the seven days of the week,

Holiday is one of several public holidays in the year,

Time of day is divided here in 48 parts each consisting of 30 minutes.

The current electricity load is included in the input indicators to the system as the predicted electricity load is highly dependent on the current rate as there is only likely to be a fluctuation in the electricity load from current electricity load. The related indicators (inputs) are grouped together because of the common connection and relation among them such as temperature, time of day etc. These groups are as follows

Predicted Temperature Group - This group contains Electricity Load, Predicted Minimum Temperature, Predicted Maximum Temperature, Time of day.

Actual Temperature Group -This group contains Electricity Load, Actual Minimum Temperature, Actual Maximum Temperature, Time of day.

Season day Group -This group contains, Electricity Load, Season (a value from 1 to 4 representing each season), Day of the week (two values, one for weekdays and zero representing weekend), Public Holiday (two values, one representing a public holidays and zero representing a working day), Time of day.

Using a hierarchical fuzzy logic system structure, it is possible to overcome this problem. The three groups created for the electricity load prediction each produce a predicted electricity load. These are then fed into the next layer of the hierarchy where the final predicted electricity load is found (see Figure 4).

The total number of rules for the hierarchical fuzzy logic system is 1455. From simulation results it was found that the hierarchical fuzzy logic system is capable of making accurate predictions of the electricity load [Mohammadian, M. and Jentzsch, R. (2005)].

FUTURE TRENDS

The grouping of input parameters of the systems considered above was performed using expert knowledge. It would be interesting to use genetic algorithms to find out the relationships between the input parameters of such systems and compare the results obtained in this way with the grouping of parameters suggested by expert.

CONCLUSION

In this paper issues in the construction of a fuzzy logic system to model a complex (nonlinear) system, namely the decomposition into hierarchical/multilayered fuzzy logic sub-systems and the learning of fuzzy rules and internal parameters is considered. Whilst the decomposition into hierarchical/multi-layered fuzzy logic sub-systems reduces greatly the number of fuzzy rules to be defined and to be learnt, other issues arise such as the decomposition is not unique and that it may give rise to variables with no physical significance. For a problem with a large number of input variables, for

Figure 4. Hierarchical fuzzy logic system for electricity load prediction
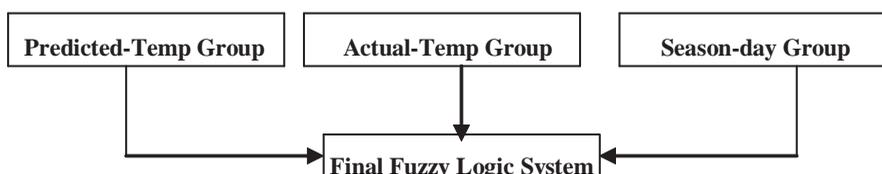
example, the problem of interest rate prediction, the non-uniqueness of the decomposition yields numerous different structures to examine in order to find one which in some sense, is the 'best' structure.

ACKNOWLEDGMENTS

REFERENCES

Bai, Y., Zhuang H. and Wang, D. (2006), Advanced Fuzzy Logic Technologies in Industrial Applications, Springer Verlag, USA, ISBN 1-84628-468-6.

Cordon, O., Herrera, F., Hoffmann, F. and Magdalena, L. (2001), *Genetic Fuzzy Systems Evolutionary Tuning and Learning of Fuzzy Knowledge Bases (Advances in Fuzzy Systems—Applications and Theory Vol. 19)*, World Scientific Publishing, USA, ISBN 981-02-4017-1.

Goldberg, D. (1989), Genetic Algorithms in Search, Optimisation and Machine Learning, Addison–Wesley, USA.

Kingham, M., Mohammadian, M, and Stonier, R. J. (1998), Prediction of Interest Rate using Neural Networks and Fuzzy Logic, *Proceedings of ISCA 7th International Conference on Intelligent Systems*, Melun, Paris, France.

Magdalena, L. (1998), Hierarchical Fuzzy Control of a Complex System using Metaknowledge, *Proceedings of the 7th International conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, France.

Mohammadian, M. (2002), Designing customised hierarchical fuzzy systems for modelling and prediction, *Proceedings of the International Conference on sinulated Evolution and Learning (SEAL'02)*, Singapore, ISBN 9810475233.

Mohammadian, M. and Kingham, M. (2004), An adaptive hierarchical fuzzy logic system for modelling of financial systems, *Journal of Intelligent Systems in Accounting, Finance and Management*, Wiley Interscience, Vol. 12, 61-82.

Mohammadian, M., Nainar, I. and Kingham, M. (1997), Supervised and Unsupervised Concept Learning by Genetic Algorithms, Second International ICSC Symposium on Fuzzy Logic and Applications ISFL'97, Zurich, Switzerland.

Mohammadian, M. and Jentzsch, R. (2005), "Electricity Load Prediction Using Hierarchical Fuzzy Logic Systems", Knowledge-Base Intelligent Information and Engineering Systems, KES2005, Springer Verlag, Australia, ISBN 3540288953.

Mohammadian, M. and Stonier, R. J. (1998), Hierarchical Fuzzy Control, *Proceedings of the 7th International conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, France.

Raju G. V. S. and Zhou, J. (1993), Adaptive Hierarchical Fuzzy Controller, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 4, 973-980, 1993.

Stonier, R. J. and Mohammadian, M. (1995), Self Learning Hierarchical Fuzzy Logic Controller in Multi-Robot Systems, *Proceedings of the IEA Conference Control95*, Melbourne Australia.

Stonier, R. J. and Mohammadian, M. (1998), Knowledge Acquisition for Target Capture, *Proceedings of the International Conference on Evolutionary Computing ICEC'98*, Anchorage, Alaska, USA.

Stonier, R. J., Stacey, A., Mohammadian, M. and Smith, S. F. (1999), Application of evolutionary learning in fuzzy logic and optimal control, *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, Vienna, Austria.

Stonier, R. J. and Zajaczkowski, J. (2003), Hierarchical fuzzy controllers for the inverted pendulum, *Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*, Singapore, ISSN 0219-613, PS01-4-03.

KEY TERMS

Fusing Variables: Fusing variables is a method for reducing the number of rules in a fuzzy rule base. The variables are fused (combined) together before input into the inference engine, thereby reducing the number of rules in the knowledge base.

Fuzzy Logic: Fuzzy sets and Fuzzy Logic were introduced in 1965 by Lotfi Zadeh as a new way to represent vagueness in applications. They are a generalisation of sets in conventional set theory. Fuzzy Logic (FL) aims at modelling imprecise models of reasoning, such as common sense reasoning for uncertain complex processes. A system for representing the meaning of lexically imprecise proposition in natural language structure through the proposition being represented as fuzzy constraints on a variable is provided. Fuzzy logic controllers have been applied to many nonlinear control systems successfully. Linguistic rather than crisp numerical rules are used to control the processes.

Fuzzy Rule Base (*Fuzzy If-Then rules):* Fuzzy If-Then or fuzzy conditional statements are expressions of the form "If **A Then **B**", where **A** and **B** are labels of fuzzy sets characterised by appropriate membership functions. Due to their concise form, fuzzy If-Then rules are often employed to capture the imprecise modes of reasoning that play an essential role in the human ability to make decision in an environment of uncertainty and imprecision. The set of If-Then rules relate to a fuzzy logic system that are stored together is called a Fuzzy Rule Base.

Genetic Algorithms: Genetic Algorithms (GAs) are algorithms that use operations found in natural genetics to guide their way through a search space and are increasingly being used in the field of optimisation. The robust nature and simple mechanics of genetic algorithms make them inviting tools for search, learning and optimization. Genetic algorithms are based on computational models of fundamental evolutionary processes such as selection, recombination and mutation.

Genetic Algorithms Components: In its simplest form, a genetic algorithm has the following components:

1. *Fitness* - A positive measure of utility, called fitness, is determined for individuals in a population. This fitness value is a quantitative measure of how well a given individual compares to others in the population.
2. *Selection* - Population individuals are assigned a number of copies in a mating pool that is used to construct a new population. The higher a population individual's fitness, the more copies in the mating pool it receives.
3. *Recombination* - Individuals from the mating pool are recombined to form new individuals, called children. A common recombination method is one-point crossover.
4. *Mutation* - Each individual is mutated with some small probability $\ll 1.0$. Mutation is a mechanism for maintaining diversity in the population.

Hierarchical Fuzzy Logic Systems: The idea of hierarchical fuzzy logic control systems is to put the input variables into a collection of low-dimensional fuzzy logic control systems, instead of creating a single high dimensional rule base for a fuzzy logic control system. Each low-dimensional fuzzy logic control system constitutes a level in the hierarchical fuzzy logic control system. Hierarchical fuzzy logic control is one approach to avoid rule explosion problem. It has the property that the number of rules needed to construct the fuzzy system increases only linearly with the number of variables in the system

Supervised Learning: A learning method in which there are two distinct phases to the operation. In the first phase each possible solution to a problem is assessed based on the input signal that is propagated through the system producing output respond. The actual respond produced is then compared with a desired response, generating error signals that are then used as a guide to solve the given problems using supervised learning algorithms.

Support Vector Machines

Cecilio Angulo
Technical University of Catalonia, Spain

Luis Gonzalez-Abril
Technical University of Catalonia, Spain

INTRODUCTION

Support Vector Machines -- SVMs -- are learning machines, originally designed for bi-classification problems, implementing the well-known **Structural Risk Minimization** (SRM) **inductive principle** to obtain good **generalization** on a limited number of learning patterns (Vapnik, 1998)**.** The optimization criterion for these machines is maximizing the margin between two classes, i.e. the distance between two parallel hyperplanes that split the vectors of each one of the two classes, since larger is the margin separating classes, smaller is the **VC dimension** of the learning machine, which theoretically ensures a good generalization performance (Vapnik, 1998), as it has been demonstrated in a number of real applications (Cristianini, 2000). In its formulation is applicable the **kernel trick**, which improves the capacity of these algorithms, learning not being directly performed in the original space of data but in a new space called feature space; for this reason this algorithm is one of the most representative of the called **Kernel Machines** (KMs).

Main theory was originally developed on the sixties and seventies by V. Vapnik and A. Chervonenkis (Vapnik et al., 1963, Vapnik et al., 1971, Vapnik, 1995, Vapnik, 1998), on the basis of a separable binary classification problem, however generalization in the use of these learning algorithms did not take place until the nineties (Boser et al., 1992). SVMs has been used thoroughly in any kind of learning problems, mainly in classification problems, although also in other problems like regression (Schölkopf et al., 2004) or clustering (Ben-Hur et al., 2001).

The fields of Optic Character Recognition (Cortes et al., 1995) and Text Categorization (Sebastiani, 2002) were the most important initial applications where SVMs were used. With the extended application of new kernels, novel applications have taken place in the field of Bioinformatics, concretely many works are related with the classification of data in Genetic Expression (Microarray Gene Expression) (Brown et al., 1997) and detecting structures between proteins and their relationship with the chains of DNA (Jaakkola et al., 2000). Other applications include image identification, voice recognition, prediction in time series, etc. A more extensive list of applications can be found in (Guyon, 2006).

BACKGROUND

Regularization Networks (RNs), obtained from the penalization inductive principle, are algorithms based on a deep theoretical background, but their purely asymptotic approximation properties and the expansion of the solution function on a large number of vectors convert them in a no practical choice in its original definition. Looking for a more reduced expansion of the solution some researchers observe the good behaviour of the SVM, being able to consider a finite training set as hypothesis in its theoretical discourse as well as building the final solution by considering nested approximation spaces.

As well regularization inductive principles as structural risk minimization establish inserting 'a priori' information on the shape of the solution without considering any assumption about the unknown probability density function relating working spaces. The regularization principle considers a regularizer or regularization operator ensuring find a good solution in asymptotic form on nested function spaces when the number of elements in the training set tends to be infinite. Besides, the SRM principle also is based on nested spaces but the solution is found by ensuring an upper bound for the risk functional considering only a finite set of empirical data.

Both inference processes are obviously not equivalents, but their similarities have been projected on the

learning methods having their theoretical background on these principles, in such a form than a number of researchers approaching the learning problem from different perspectives are implied in establishing a common framework allowing to deal SVMs and RNs like particular cases of a more general learning methodology, let us call it Kernel Methods (Campbell, 2000), when it is emphasized the key rule played by the kernel function generating the feature space, or Large Margin Classifiers (Cristianini, 2000), when the measure to be optimized to ensure maximal generalization is emphasized.

Both, results obtained and theoretical framework from SRM seem offer better theoretical warranties that other previous approaches when looking for solutions with good generalization for problems based on a finite empirical set. Hence, the integration of machine learning methods on mixed models is a state of the art research field. Besides, the use of Bayesian inference in these mixed models is being avoided because the user must beforehand define a probability density function.

SUPPORT VECTOR MACHINE

Let us consider a bi-classification problem (another kind of problems are analyzed in the cited references). Thus, let $Z=\{z_i=(x_i,y_i), i=1,2,...,n\}$ be a training set with $x_i \in X \subset R^d$ as the input space and $y_i \in \{\theta_1, \theta_2\}$ (the output space) ($\theta_1 \neq \theta_2$). Let us initially suppose that classes are linearly separable (two sets are linearly separable in n-dimensional space if they can be separated by an $d-1$ dimensional hyperplane) then a hyperplane, denoted by $\pi : w\,x - b = 0$ (where b is called bias), is sought which separates the two classes, that is $w\,x_i - b > 0$ if $y_i = \theta_1$ and $w\,x_i - b < 0$ if $y_i = \theta_2$. Nevertheless, there are many hyperplanes with this condition (see **Figure 1**), so a new condition is imposed that is the distance between the optimal hyperplane and the nearest training pattern (margin) is maximal. Let us see detailed this condition: In the first place without lost of majority let us suppose that $\theta_1 = 1$ and $\theta_2 = -1$. Hence, let β and α be the minimum (class $+1$) and the maximum (class -1) absolute values of the unbiased hyperplane effectively attained for some patterns $z_1 \in Z_1$ and $z_2 \in Z_2$ i.e.

$$\alpha = \max_{z_i \in Z_2} w\,x_i \text{ and } \beta = \min_{z_i \in Z_1} w\,x_i,$$

where Z_1 and Z_2 are the patterns belonging to the classes labelled as $\{+1,-1\}$ respectively. It is considered that $\alpha \leq \beta$, otherwise vector $-w$ is chosen. Thus, given a vector w, the margin is defined as the distance between parallel hyperplanes $\pi_\alpha : w\,x - \alpha = 0$ and $\pi_\beta : w\,x - \beta = 0$, that is

$$margin = d(\pi_\alpha, \pi_\beta) = \frac{\beta - \alpha}{\|w\|}$$

Figure 1. Type A denotes the class +1 (θ_1) and Type B denotes the class -1 (θ_2).
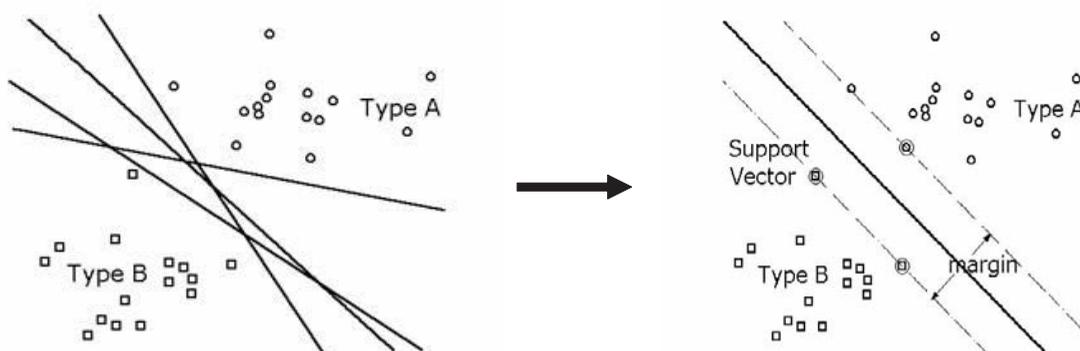
(see **Figure 1**). The natural choice for the bias, ensuring positive and negative outputs for the patterns in the respective classes, is

$$b = \frac{\alpha + \beta}{2}.$$

The maximization of the margin has the objective to force the generalization of the found learning machine (Vapnik, 1995, Schölkopf et al., 2002).

The extension to non-lineal functions of decision is carried out introducing the input space $X \subset R^d$ in another space, usually with higher dimension F, called feature or characteristics space which is endowed with an inner product, through a non-lineal injection, $\phi : X \rightarrow F$ (this procedure is called kernel trick), such that the optimal hyperplane

$$f(x, w) = \langle \varphi(x), w \rangle_F - b$$

is sought in the feature space F. Nevertheless, with the objective of defining in a unique way the searched hyperplane (canonical form) next restrictions should be added:

$$y_i f(x_i) \geq 1 - \xi_i \quad i = 1, 2,..., n$$

on the training set Z, where the slack variables $\xi_i \geq 0$ are introduced to allow that some examples exist violating the constraint imposed by the margin (soft-margin) because it should be considered the possibility that the classes to be separated are overlapped or that patterns contain noise that is the set Z can be a non-separable linearly. Hence, the function $f(x,w)$ allows defining the decision function as

$$h(x) = sign(f(x, w))$$

that is, given a new input x the label assigned by the machine is θ_1 if $h(x)=1$ and θ_2 otherwise.

Thus the optimal hyperplane accomplishes the following problem of constrained optimization:

$$Min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.a. \quad y_i f(x_i) \geq 1 - \xi_i \quad i = 1,2,\cdots,n$$
$$\xi_i \geq 0 \quad i = 1,2,\cdots,n$$

The solution vector can be written as

$$w = \sum_{i=1}^{SV}\alpha_i y_i \phi(x_i) \tag{1}$$

where SV is the number of training vectors which verify that their corresponding Lagrange multiplier α_i is no null (these vectors are called support vectors) (see **Figure 1**). Many other different approaches for defining SVM exist (González et al., 2006), nevertheless this formulation is the most usual.

From the equation (1), the optimal hyperplane can be written as:

$$f(x) = \sum_{i=1}^{SV}\alpha_i y_i k(x_i, x) - b$$

where $k(x_i, x) = \langle \phi(x_i), \phi(x)\rangle_F$ is a **Kernel** (a bivariate function accomplishing the Mercer's theorem) and b is calculated by using the Karush-Kuhn-Tucker (KKT) conditions.

For multi-classification problems, a set of possible labels $Y = \{\theta_1, ..., \theta_\ell\}$ with $\ell \geq 2$ is considered. There are two main SVM-based approaches to solve these problems. A first one is the "all the classes at once", which solves these problems by considering all the instances from all the classes in a unique optimization formulation, whereas the other one is the "decomposition-reconstruction" architecture approach (multi-classification in two phases), using binary SVMs.

In the first case, several formulations exist (Vapnik, 1998, Cramer, 2001, Aiolli, 2005), however among all of the proposed approaches to the maximal margin problem, that presented in Shashua et al. (2002) is the only one considering to maximize the exact expression of the margin between instances with different label, so the multi-classification problem is interpreted like an ordinal regression problem where the objective function is the sum of the inverse of the margins between classes.

In the case of multi-classification in two phases, the most usual multi-classification SVM approaches are 1-v-1 (one-versus-one) SVM and 1-v-r (one-versus-rest) SVM. In both approaches, a first decomposition phase generates several learning machines in parallel and a reconstruction scheme allows obtaining the overall output by merging outputs from the decomposition phase.

In the first phase of 1-v-r SVM, each machine takes in consideration all the classes; ℓ binary classifiers are trained to generate hyperplanes f_k, $(k = 1,2,...,\ell)$ separating training vectors with label θ_k from the remaining vectors. In the reconstruction phase (second phase), a labels distribution generated by the trained machines in the parallel decomposition is considered through a merging scheme. All the information provided by the training vectors is considered, main drawback being that it is not well designed to separate specific classes.

In the first phase of 1-v-1 SVM, each machine takes in consideration only two classes. In this approach,

$$\frac{\ell(\ell-1)}{2}$$

binary classifiers are trained to generate hyperplanes f_{kh}, $k, h = 1,2,...,\ell, k<h$ separating training vectors with label θ_k from training vectors in class θ_h. Remaining training vectors are not considered in the optimization problem. In the reconstruction phase, a labels distribution generated by the trained machines in the parallel decomposition is considered through a merging scheme. Main drawback is that only data from two classes are considered for each machine in the decomposition procedure so output variance is high and any information from the rest of classes is ignored.

The 1-v-1 scheme is usually preferred because it takes less training time (Kressel, 1999} although some researches consider the 1-v-r scheme since this scheme has some advantages (Rifkin et al., 2004). Nevertheless according to (HsuLin2002) it would be difficult to say which one gives better accuracy.

FUTURE TRENDS AND CONCLUSION

A recursive problem when considering SVM is to reduce the computational cost when the QP problem is being solved.

SVM for classification is the most studied approach; however other problems as regression are not enough developed to be competitive versus other standardized research areas, like artificial neural networks. So, a long way still remains to be walked in this area.

A particular open research problem in the case of multi-classification is the implementation of the tri-class scheme. In this approach, one class is label as +1, another class as -1 and the rest of the classes as 0 which is forced to be encapsulated into a δ-tube, $0<\delta<1$, along the separation hyperplanes. The tri-class SVM improves standard algorithms treating 2-class classification problems during the decomposing phase of a general multi-class scheme by focusing the learning on

Figure 2. The circles are the class +1, the rectangles are the class 0 and the triangles are the class -1.
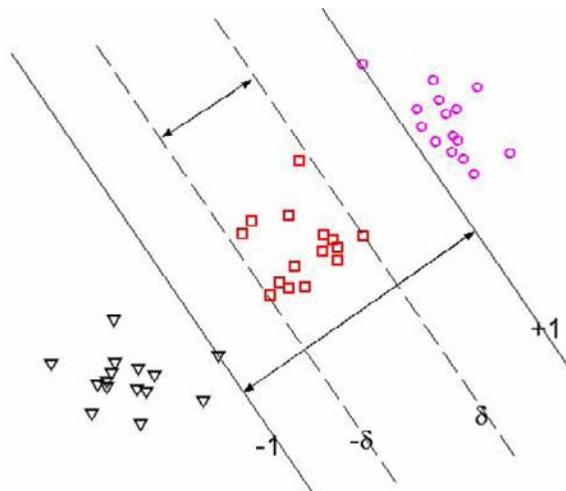
two classes, but using all the available information on the patterns (see **Figure 2**), so this approach can be seen as a mixed between the 1-v-r and 1-v-1 SVM. A second theoretical advantage of the "third-class approach" is the robustness of the reconstruction procedure (Angulo et al., 2003), which could drive to empirically expect a higher performance of the new approach in terms of accuracy (Angulo et al., 2006). Research should even include the study of theoretical generalization bounds for this kind of machine.

REFERENCES

Angulo, C. & Parra, X. & Catalá, A. (2003). K-SVCR. A Support Vector Machine for Multi-Class Classification. *Neurocomputing*, 55(1-2), 57-77.

Angulo, C. & Ruiz, F. & González, L. & Ortega, J.A. (2006). Multi-classification by using Tri-class SVM. *Neural Processing Letters*, 23(1), 89-101.

Aiolli, F. & Sperduti, A. (2005). Multiclass Classification with Multi-Prototype Support Vector Machine. *Journal of Machine Learning Research*, 6, 817—850.

Ben Hur, A. & Horn, D. & Siegelmann, H. & Vapnik, V. (2001). Support Vector Clustering. *Journal of Machine Learning Research*, 2, 125-137.

Boser, B.E. & Guyon, I. & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 144-152.

Brown, M. & Grundy, W. & Lin, D. & Cristianini, N. & Sugnet, C. & Furey, T. & Ares, M. & Haussler, D. (1997). Knowledge-based Analysis of Microarray Gene Expression Data Using Support Vector Machines. *Proceedings of the National Academy of Sciences*, 97(1), 262-267.

Campbell, C. (2000). An introduction to kernel methods. In Howlett, R. and Jain, L. editors, *Radial Basis Function Networks: Design and Applications*, Berlin, Springer Verlag.

Cortes, C. & Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20, 273-297.

Crammer, K. & Singer, Y. (2001). On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research*, 2, 265-292.

Cristianini, N. & Shawe-Taylor, J. (2000). An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press.

Hsu, C. & Lin, C. (2002). A comparison of methods for multiclass support vector machine. *IEEE Transactions on Neural Networks*, 13(2), 415-425.

González, L. & Angulo, C. & Velasco, F. & Catala, A. (2006). Dual unification of bi-class Support Vector Machine formulations. *Pattern Recognition*, 39(7), 1325-1332.

Guyon, I. (2006). SVM application list. http://www.clopinet.com/isabelle/projects/svm/applist.html

Jaakkola, T. & Diekhans, M. & Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2), 95-114.

Kressel, U. (1999). Pairwise classification and support vector machine. Advances in Kernel Methods: Support Vector Learning. MIT Press. Cambridge, MA, 255-268.

Rifkin, R. & Klautau, A. (2004). In defense of one-vs-all classification, *Journal of Machine Learning Research*, 5, 101-141.

Shashua, A. & Levin, A. (2002). Taxonomy of Large Margin Principle Algorithms for Ordinal Regression Problems. *Neural Information Processing Systems*, 16.

Schölkopf, B. & Smola, A.J. (2002). Learning with Kernels. The MIT Press. Cambridge, MA.

Schölkopf, B. & Smola, A.J. (2004). A Tutorial on Support Vector Regression, *Statistics and Computing*, 14, 199-222.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1-47.

Vapnik, V. (1995). The nature of statistical learning theory. Springer. New York.

Vapnik, V. (1998). Statistical Learning Theory. John Wiley & Sons, Inc.

Vapnik, V. & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16, 264-280.

Vapnik, V. & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24.

KEY TERMS

Generalization: It is the process of formulating general concepts by abstracting common properties of instances. In the context-specific of the SVMs means that if a decision function $h(x)$ is obtained from $x_i \in X \subset R^d$, this function is considered valid for all $x \in R^d$. It is the basis of all valid deductive inference and a process of verification is necessary to determine whether a generalization holds true for any new given instance.

Kernel Machine or Kernel Methods: Kernel machine owe their name to the use of kernel functions that enable them to operate in the feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. Kernel functions have been introduced for sequence data, text, images, as well as vectors (Schölkopf et al., 2002).

Kernel Trick: This procedure consists on substituting the inner product in the space of input variables by an appropriate function $k(x,x')$ that associates to each two inputs an real number real that is $k(x,x') \in R$, for any $x,x' \in X$. Thus, this is a method for converting a linear classifier algorithm into a non-linear one by using a non-linear function to map the original observations into a higher-dimensional space; this makes a linear classification in the new space equivalent to non-linear classification in the original space.

Mercer's Kernel: A function $k : X \times X \to R$ is a Mercer's kernel if it is a continuous, symmetric and positive semi-definite function (Cristianini, 2000).

Regularization: It is any method of preventing overfitting of data by a model and it is used for solving ill-conditioned parameter-estimation problems.

Structural Risk Minimization (SRM) Inductive Principle: The main idea of the principle is to minimize a test error by controlling two contradictory factors: a risk functional from empirical data and a capacity for the set of real-valued functions (Vapnik, 1998).

The VC Dimension (for **Vapnik-Chervonenkis Dimension**)**:** It is a number which is defined as the cardinality of the largest set of points that an algorithm can shatter, that is it is a measure of the capacity of a statistical classification algorithm (Vapnik et al., 1971).

A Survey on Neural Networks in Automated Negotiations

Ioannis Papaioannou
National Technical University of Athens, Greece

Ioanna Roussaki
National Technical University of Athens, Greece

Miltiades Anagnostou
National Technical University of Athens, Greece

INTRODUCTION

Automated negotiation is a very challenging research field that is gaining momentum in the e-business domain. There are three main categories of automated negotiations, classified according to the participating agent cardinality and the nature of their interaction (Jennings, Faratin, Lomuscio, Parsons, Sierra, & Wooldridge, 2001): the bilateral, where each agent negotiates with a single opponent, the multi-lateral which involves many providers and clients in an auction-like framework and the argumentation/persuasion-based models where the involving parties use more sophisticated arguments to establish an agreement. In all these automated negotiation domains, several research efforts have focused on predicting the behaviour of negotiating agents. This work can be classified in two main categories. The first is based on techniques that require strong a-priori knowledge concerning the behaviour of the opponent agent in previous negotiation threads. The second uses mechanisms that perform well in single-instance negotiations, where no historical data about the past negotiating behaviour of the opponent agent is available. One quite popular tool that can support the latter case is Neural Networks (NNs) (Haykin, 1999).

NNs are often used in various real world applications where the estimation or modelling of a function or system is required. In the automated negotiations domain, their usage aims mainly to enhance the performance of negotiating agents in predicting their opponents' behaviour and thus, achieve better overall results on their behalf. This paper provides a survey of the most popular automated negotiation approaches that

are using NNs to estimate elements of the opponent's behaviour.

The rest of this paper is structure as follows. The second section elaborates on the state of the art bilateral negotiation frameworks that are based on NNs. The third section briefly presents the multilateral negotiation solutions that exploit NNs. Finally, in the last section a brief discussion on the survey is provided.

NEURAL NETWORKS IN BILATERAL NEGOTIATIONS

In (Zhang, Ye, Makedon, & Ford, 2004) a hybrid bilateral negotiation strategy mechanism is described that supplies negotiation agents with more flexibility and robustness in an automated negotiation system. The framework supports a dynamically assignment of an appropriate negotiation strategy to an agent according to the current environment, along with a mechanism to create new negotiation rules by learning from past negotiations. These learning capabilities are based on feedforward back-propagation neural networks and multidimensional inter-transaction association rules. However, the framework is not adequately described and defined and the neural networks are not specifically instantiated. Additionally, there are neither quantitative nor qualitative experimental results for real world cases. Finally, the format of the input to the generic network that is presented is ambiguously described.

In (Zeng, Meng, & Zeng, 2005), the authors employ a neural network to assist the negotiation over very specific issues from a real world example. The network is trained online by the past offers made by the op-

ponent, while both the buyer and the seller agent have the ability to employ the proposed network. However, the experimental data sets are very restrictive and do not address the diversity of those that can be arisen in real scenarios. Additionally, the authors do not present the actual size of the hidden layer, a parameter that is extremely crucial with regards to the appropriateness to use such a network in a real time negotiation procedure by an agent with limited resources.

Furthermore, in (Rau, Tsai, Chen, & Shiang, 2006), the authors studied the negotiation process between a shipper and a forwarded using a learning-based approach, which employed a feedforward back-propagation neural network with two input data models and the negotiation decision functions. Issues of the negotiation were the shipping price, delay penalty, due date, and shipping quantity. The proposed mechanism was applicable to both parties at the same time and the network architecture was chosen based on past similar attempts, following a very restrictive pattern for the number of the hidden layer's neurons. The conducted experiments showed an overall improvement of the results for both negotiating parties, while the framework was proven stable and with small deadlock probability. However, as its authors support, further experimentation is required especially with regards to a wider variety of strategies and possibly more suitable network architectures for the hidden layer.

In (Carbonneau, Kersten, & Vahidov, 2006), a neural network based model is presented for predicting the opponent's offers during the negotiation process. The framework was tested over a specific set of experimental data collected from other existent frameworks and it is highly adjusted to these data. The purpose of this solution is not only to predict the opponent's next offer, but also the perception for the specific procedure, i.e. an overall vision on why everything is happening and where the procedure is led. Thus, the prediction of the opponent's next round offer is only a part of the network's output. However, the chosen experiment set is constrained and doesn't examine the effectiveness of the framework on diverse strategies as those proposed in the very first steps of the area and are now mainly used (Faratin, Sierra, & Jennings, 1998). Additionally, although the authors support the view that their framework is proper for real-time environments, the fact is that the resulted network is difficult to be online trained, mainly because of its size and the resources that are required for such training. Thus, this network

architecture is probably inappropriate for mobile agents' environments, and something smaller and more specific should be designed, due to the limitations that these environments share.

Moreover, in (Oprea, 2003), the author presents a shopping agent, which is capable of negotiating in online bilateral, multi-issue procedures using an offline created and trained feedforward neural network in order to increase its profitability by adapting its behaviour according to its opponent's. The purpose of the neural network's application on each procedure is to predict the opponent's next offer on a round by round basis and thus, model its behaviour and intentions in order to finally achieve a better or even the best possible deal. With the exploitation of the neural network the shopping agent can decide during the online phase of negotiation, which is the opponent's strategy and estimate its reservation value. Concerning the experiments conducted, the author uses the well-justified negotiation tactics presented in (Faratin, Sierra, & Jennings, 1998) in order to test the proposed solution and concludes that the framework is working well in case of medium or long term agents' deadlines. However, the results presented are not thoroughly justified and more extreme opponent strategies should be tested in order to decide on the network's adequacy for such environments. Probably, the three hidden layer neurons might not be sufficient for such cases and long-term estimations.

Finally, Papaioannou et al. have recently designed and evaluated several single-issue bilateral negotiation approaches, where the Client agent is enhanced with Neural Networks. More specifically, in (Roussaki, Papaioannou, & Anagnostou, 2006), the Client agent uses a lightweight feedforward back-propagation NN coupled with a fair relative tit-for-tat imitative tactic, and attempts to estimate the Provider's price offer upon the expiration of the Client's deadline. This approach increases the number of agreements reached by one third in average. In (Papaioannou, Roussaki, & Anagnostou, 2006), the performance of MLP and RBF NNs towards the prediction of the Provider's offers at the last round has been compared. The experiments indicate that the number of agreements is increased by ~38% in average via both the MLP- and the RBF-assisted strategies. Nevertheless, the overall time and the number of neurons required by the MLP are considerably higher than these required by the RBF. In (Roussaki, Papaioannou, & Anagnostou, 2007), MLP and GR NNs have been used by the Client agent in order to identify the unsuccess-

ful negotiation threads (UNTs) at an early stage, thus terminating them long before the deadlines expire. It has been observed that the MLP NN detects more than 90% of UNTs in average, outperforming by little the GR NN. Finally, in (Papaioannou, Roussaki, & Anagnostou, 2007), the performance of MLP and RBF NNs has been compared with cubic splines, least-square-based polynomial approximators, exponential approximators and Gaussian approximators, in order to predict the future offers of the negotiating Provider Agent. The wide experimental evaluation conducted indicates that both the MLP- and the RBF-assisted negotiation strategies perform almost equally well and outperform the other four approximator-assisted strategies. In this paper, the proposed framework is extended to address multi-issue negotiations considering the significance of the issues under negotiation for the negotiating party, as well as their degree of interdependency. A disadvantage in the aforementioned NN-based negotiation frameworks is that they have only been evaluated in case the Provider agent adopts a time-dependent strategy.

NEURAL NETWORKS IN MULTILATERAL NEGOTIATIONS

In (Oprea, 2001), the use of a small-scaled feedforward neural network is attempted in order to predict the opponent agent's behaviour. In this framework the enhanced agent is negotiating against an opponent that is not equipped with any learning or other intelligent mechanism. The neural network is properly constructed and trained at every round to respond with the opponent's next value at each negotiation step using only the three prior offers issued by the opponent. This fact makes the step-by-step computation feasible in real time procedures, but not necessarily reliable. However, the proposed approach was proved adequate only in cases when either both agents (or at least the opponent agent) have long-term deadlines.

A different usage of the neural networks' potential is presented in (Shibata, & Ito, 1999), where the authors are mainly concerned with the communication between agents. In principal, they divide the agents' communication into two classes with respect to its meaning. The first one incorporates the cases where the agent transmits the observed information while the second those where the agent's intention is transmitted. The framework exploits an Elman recurrent neural network

with feedback loops, especially for the latter class of cases. The network assists the agents to avoid possible negotiation deadlocks, although nothing is known apriori with regards to their strategy or resources. The network keeps the past information and adapts online its corresponding agent's behaviour accordingly in order to avoid collisions. The proposed framework was also tested with four agents leading to promising results. However, the authors don't propose or apply techniques for higher profitability of the participating agents but only for collision avoidance by learning the opponent's intention. Additionally, a recurrent neural network is a complex structure and seems inappropriate for application in low resources agent environments.

Furthermore, in (Abreu, Canuto, & Santana, 2005) the authors present a comparative analysis of some negotiation methods used in a multi-neural agent system, called NeurAge. This system is composed of several neural classifiers, called neural agents, and its main aim is to overcome some drawbacks of multi-classifier systems and, as a consequence, to improve their performance. These neural agents provide a common output, which results after negotiation among them and it is the system's output. For this purpose, three different negotiation methods are evaluated: the game theoretic, the auction based and the confidence based ones. The results prove that the proposed approach is valuable for such classifier systems and might end up being valuable in cases where tactic classification should be conducted. However, the system is inappropriate for online procedures, requires cooperation between multiple neural agents and has not been tested on real negotiation tactics' numerical data. Therefore, the results might be valuable when a classification scheme is required, but are probably inappropriate as a future prediction pattern.

On the other hand, (Veit, & Czernohous, 2003) present the results of enhancing consumer agents with several machine-learning algorithms in a properly designed electronic market with one static supplier. The results prove that under very specific circumstances the neural network assisted agent performs worse than a simple Q-learning assisted agent that maintains a specific set of values for the learning procedure in an a-priori instantiated matrix. However, the scenarios are very restrictive and in no case address the characteristics of real world ones where the application of similar table based agents would fail mainly due to the diversity of the potential solution spaces for each

negotiation. Besides, the authors themselves admit this remark, including it in their future plans.

In (Park, & Yang, 2006), the authors propose a negotiation agents system based on the incremental learning of a feedforward neural network in order to increase the efficiency of bilateral negotiations and to improve the applicability towards multilateral negotiations. The network is triggered with values that are extracted after a utility evaluation procedure and at each round the output is forming the next counter-offer of the party. With regards to the generalization to the multilateral case, the proposed approach is based on matching all sellers and all buyers in pairs among all possible ones, following practical criteria as the common negotiation range term used, indicates. The experimental results show that the proposed system achieves up to 2% more agreements and carries out the negotiations at least twice as fast as others with similar settings.

In (Wang, Chai, & Huang, 2005), the authors attempt to solve the problem of selecting a selling agent that meets buyer user's requirements as well as his utility constraints as those represented by the corresponding intelligent agent. The problem is solved by choosing the seller before the negotiation and thus, the accuracy of the negotiation and the buyer's utility are improved. In order to fully utilize negotiation history, this paper transforms the problem of choosing seller into a K-armed bandit problem. The utility function is a joint summation of the utilities of both the buyers and the sellers, while the buyer uses a properly learned neural network in order to learn its opponents' preferences and finally choose the one that will lead to the best agreement. The advantage of this framework is that the buyer's neural network learns off-line and only uses the results for the online procedure. Thus, there is not substantial impact on the real procedure.

Finally, in (Liu, & You, 2003), a fuzzy neural network is proposed to deal with the uncertainties in real world shopping activities, such as consumer preferences, product specification, product selection, price negotiation, purchase, delivery, after-sales service and evaluation. The fuzzy neural network manages to achieve an automatic and autonomous product classification and selection scheme to support fuzzy decision-making by integrating fuzzy logic technology and the back-propagation feedforward neural network. In addition, a visual data model is introduced to overcome the limitations of the current web browsers that lack flexibility for customers to view products from different perspectives. The experimental results demonstrate the feasibility of the proposed approach for web-based business transactions.

CONCLUSION AND DISCUSSION

In this paper, a brief survey of the most popular research efforts in the field of NN-assisted automated negotiations is presented. An important observation that can easily be made is that that there is a substantial diversity on the purposes that the NNs are used for in this domain. For instance, in some cases they aim to estimate the opponent's future offers, whereas in other cases they assist the negotiating agent on selecting the best tactic that should be used in order to increase its potential utility. Even though the usage of NNs in automated negotiations may enhance various aspects of their performance and results, there are some cases where they are not suitable. For example, they perform far better when they are trained off-line, thus being less suitable when no a-priori knowledge is available. In general, it is preferable that relatively small NNs that are trained off-line are used, but if this is not possible, it is better to use NNs of minimal size that are trained on-line, risking however that they will eventually not be suitable enough. Furthermore, if the negotiation strategy of the opponent is not consistent, thus frequently demonstrating sharp changes in the type or configuration of the tactic used, the NNs often fail to adjust. In case the opponent employs imitative negotiation strategies, the usability of NNs in estimating the opponent's behaviour is questionable. Finally, if the agent has low storage and processing resources available, the NNs that can be employed need to be so lightweight that they considerably lack flexibility. Despite these shortcomings, it is expected that NNs will gain a considerable share in the learning-enabled negotiating agents in the electronic marketplace.

REFERENCES

Abreu, M., Canuto, A., & Santana, L. (2005). A Comparative Analysis of Negotiation Methods for a Multi-neural Agent System. *5th International Conference on Hybrid Intelligent Systems* (HIS 2005), Rio de Janeiro, Brazil.

Carbonneau, R., Kersten, G., & Vahidov, R. (2006). Predicting Opponent's Moves in Electronic Negotiations Using Neural Networks. *International Conference of Group Decision and Negotiation* (GDN 2006), Karlsruhe, Germany.

Faratin, P., Sierra, C., & Jennings, N. (1998). Negotiation Decision Functions for Autonomous Agents. *International Journal of Robotics and Autonomous Systems*. (24)3-4, 159-182.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd edition). London UK: Prentice Hall.

Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., & Wooldridge, M. (2001). Automated Negotiation: Prospects, Methods, and Challenges. *International Journal of Group Decision and Negotiation*. (10)2, 199-215.

Liu, J., & You, J. (2003). Smart Shopper: An Agent-Based Web-Mining Approach to Internet Shopping. *IEEE Transactions on Fuzzy Systems*. (11)2, 226-237.

Oprea, M. (2001). Adaptability and Embodiment in Agent-Based Ecommerce Negotiation. *Workshop Adaptability and Embodiment Using Multi-Agent Systems* (AEMAS 2001), Prague, Czech Republic.

Oprea, M. (2003). The Use of Adaptive Negotiation in Agent-Mediated Electronic Commerce. *Lecture Notes on Artificial Intelligence* (LNAI). Springer-Verlag, Berlin Heidelberg New York. 2691, 594-605.

Papaioannou, I., Roussaki, I., & Anagnostou, M. (2006). Comparing the Performance of MLP and RBF Neural Networks Employed by Negotiating Intelligent Agents. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (IAT 2006), Hong Kong, China.

Papaioannou, I., Roussaki, I., & Anagnostou, M. (2007). Comparing Polynomial Approximators to Neural Networks for Agent Behaviour Prediction in e-Negotiations, submitted for publication to the *ACM Transactions of Autonomous and Adaptive Systems*.

Park, S., & Yang, S. (2006). An Automated System based on Incremental Learning with Applicability Toward Multilateral Negotiations. *International Joint Conference SICE-ICASE*, Busan, Korea.

Rau, H., Tsai, M., Chen, C., & Shiang, W. (2006). Learning-based automated negotiation between shipper and forwarder. *Journal of Computers and Industrial Engineering*, (51)3, 464-481.

Roussaki, I., Papaioannou, I., & Anagnostou, M. (2006). Employing Neural Networks to Assist Negotiating Intelligent Agents. *2nd IEE International Conference on Intelligent Environments 2006* (IE 2006), Athens, Greece.

Roussaki, I., Papaioannou, I., & Anagnostou, M. (2007). Building Automated Negotiation Strategies Enhanced by MLP and GR Neural Networks for Opponent Agent Behaviour Prognosis. *Lecture Notes of Computer Science* (LNCS). Springer-Verlag, Berlin Heidelberg New York. 4507, 152-161.

Shibata, K., & Ito, K. (1999). Emergence of Communication for Negotiation By a Recurrent Neural Network. *4th International Symposium on Autonomous Decentralized Systems*, Tokyo, Japan.

Veit, D., & Czernohous, C. (2003). Automated Bidding Strategy Adaptation using Learning Agents in Many-to-Many e-Markets. *2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS 2003), Melbourne, Australia.

Wang, L.M., Chai, Y.M., & Huang, H.K. (2005). Choosing optimal seller based on off-line learning negotiation history and k-armed bandit problem. *International Conference on Machine Learning and Cybernetics* (ICMLC 2005), Guangzhou, China.

Zeng, Z.M., Meng, B., & Zeng, Y.Y. (2005). An Adaptive Learning Method in Automated Negotiation. *International Conference on Machine Learning and Cybernetics* (ICMLC 2005), Guangzhou, China.

Zhang, S., Ye, S., Makedon, F., & Ford, J. (2004). A Hybrid Negotiation Strategy Mechanism in an Automated Negotiation System. *5th ACM Conference on Electronic Commerce* (EC 2004), New York, USA.

KEY TERMS

Automated Negotiation: It is the process by which group of actors communicate with one another aiming to reach to a mutually acceptable agreement on some matter, where at least one of the actors is an autonomous software agent.

Bilateral Negotiation: A negotiation procedure, where exactly two parties are involved, i.e. a client and a provider.

Multilateral Negotiation: A negotiation procedure, where more than two parties are involved, i.e. multiple clients and/or providers negotiate simultaneously.

Multi-Layer Perceptron (MLP): A fully connected feedforward NN with at least one hidden layer that is trained using back-propagation algorithmic techniques.

Neural Network (NN): A network modelled after the neurons in a biological nervous system with multiple synapses and layers. It is designed as an interconnected system of processing elements organized in a layered parallel architecture. These elements are called neurons and have a limited number of inputs and outputs. NNs can be trained to find nonlinear relationships in data, enabling specific input sets to lead to given target outputs.

Radial Basis Function (RBF): Function that involves a distance criterion with respect to a centre, such as a circle, ellipse or Gaussian.

RBF NN: It is an artificial NN, the activation functions of which are radial basis functions. It has two layers of processing, where the first maps the input onto each RBF neuron in the other (hidden) layer.

Swarm Intelligence Approach for Ad–Hoc Networks

Prayag Narula
University of Delhi, India

Sudip Misra
Yale University, USA

Sanjay Kumar Dhurandher
University of Delhi, India

INTRODUCTION

Wireless ad-hoc networks are infrastructureless and they consist of nodes that come together and start communicating dynamically without requiring any backbone support. The nodes can enter and leave the network at will and can move about in the network at will.

Ad-hoc networks present the perfect test-beds for bio-inspired computing algorithms. Both ad-hoc networks and bio-inspired computing approaches are characterized by self-organization, feedback and structural and functional complexity (Toh, 2002) (deCastro & Von Zuben, 2005). Hence, bio-inspired algorithms often provide us an opportunity to solve the most complex problems of ad-hoc networks in a satisfactory manner. In this chapter, we present the works done in the field of ad-hoc networks using bio-inspired Swarm Intelligence (SI). In particular, we look at how we can use Ant Colony Optimization (ACO) technique, a SI technique, for optimal routing in ad-hoc networks.

BACKGROUND

Most major bio-inspired algorithms have found implementations in the field of ad-hoc networks. Before delving into the details of the applications of ACO techniques for solving problems in ad-hoc networks, for contextual alignment, let us broadly review some of the applications of the different classes of bio-inspired algorithms to ad-hoc networks. Barolli, Koyama, & Shiratori (2003) presented a Genetic Algorithm to solve QoS routing for ad-hoc networks, while Di Caro, Ducatelle, & Gambardella (2004) used ACO technique to develop a nature inspired routing algorithm for ad-hoc

networks. On similar lines, Wedde & Farooq (2005) presented BeeAdHoc – a routing algorithm inspired from foraging behavior of honey-bees. Neural Networks too have been extensively used in ad-hoc networks for providing solution to the problems of routing (Vicente, Mujica, Sisalem, & Popescu-Zeletin, 2005), intrusion detection (Zhang & Lee, 2000) and clustering (Ai-bin, Zi-xing, & De-wen, 1993).

ACO is one of the most popular techniques among different bio-inspired techniques and has been extensively studied and deployed for solving problems as varied as Vehicular Routing Problem (VRP) (Toth & Vigo, 2001) to Single Machine Total Weighted Tardiness Problem (SMTWTP) (Abdul-Razaq, Potts & Van Wassenhove, 1990) to Graph Colouring Problem (Vesel & Zerovnik, 2000).

ACO was introduced by Marco Dorigo in his PhD. thesis as Ant System (AS). It was initially aimed at solving the popular Travelling Salesperson's problem. Though the solution provided by AS was suboptimal when compared with other specialized solutions, it underlined a method which models the foraging behaviour of ants to solve complex problems of computer science.

MAIN FOCUS OF THE CHAPTER

As mentioned earlier, the primary focus of this Chapter is to illustrate the applications of bio-inspired ACO techniques in the field of ad-hoc networks. We start by introducing the concepts of SI. Then, ACO concepts and their implementations in ad-hoc networks are discussed in detail. We first present and explain properties of ant colony that can enable to find the shortest path between

the source of the food of the ants and their colony using the concept of *pheromone*. We explain the concept of artificial ants and then present the *Random Proportional Transition Rule* (Dorigo & Stützle, 2003). We, then, describe in detail, the *AntHocNet* algorithm (Di Caro, Ducatelle, & Gambardella, 2004), which uses the ACO technique for routing data in ad-hoc networks.

SWARM INTELLIGENCE

Social insects such as ants, bees, wasps and termites and organisms such as fishes and birds rely on local communication to achieve distributed control. While insects such as ants, bees and termites rely on indirect communication through environment (also often referred to in the literature as *Stigmergy*), birds are dependent on direct but localised communication.

Nonetheless, all of these techniques aim at developing a system in which each element of the system works together to establish autonomy. The elements co-operate with each other locally to make the system much more adaptable and robust to changes and errors. Since these are the main aims of the design of ad-hoc networks, SI algorithms are effectively employed for

solving routing and Quality-of-Service (QoS) problems in ad-hoc networks.

ACO

Natural ants have a property that they always find the shortest path to the food source from their nests. This property can be illustrated by the experiments explained in (Goss, Aron, Deneubourg, & Pasteels, 1989) and (Deneubourg, S., S., & J., 1990). The set up of the experiments is illustrated in Figure 1 and Figure 2.

In Figure 1, the path between the nest and the food source are equal. It was found that roughly 50% of the ants were using each path. On the other hand in the set up shown in Figure 2, when the paths are unequal, it was found that after some time nearly all the ants were using the smaller path. This phenomenon can be explained using the following argument.

It was found that the ants mark the path that they take by a chemical named *pheromone*, thereby guiding other ants to take that path. Its implication is that the ants choose a path on the basis of the amount of pheromone lying on that path. In Figure 2, when the first group of ants start from their nest, they choose each path with equal probability. So, about half of the ants start moving on each path. The ants using the smaller

Figure 1. Same path lengths[1]
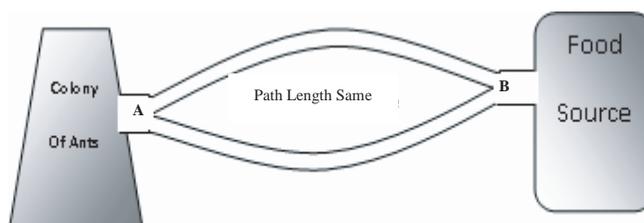
Figure 2. Different path lengths[2]
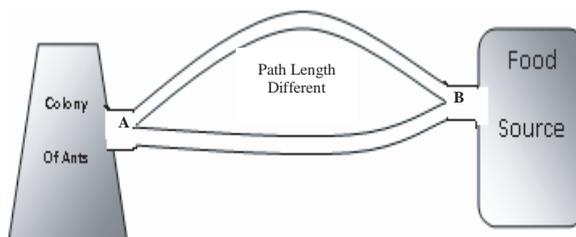
path reach point B first. Since no pheromone is present, the group again divides into two with equal number of ants turning towards the nest and other half going back via the longer path. When the ants coming from the longer path (which remain unperturbed by the ants coming back from that path) reach point B, they face a choice between going towards the food source or going back towards point A using the smaller path. Since pheromone on path to the food source is smaller than that on the path going back to point A, more ants use this path, thereby increasing the pheromone quantity on shorter path even further. When the second group of ants start from the nest and reach point A, more ants chose the smaller path since the pheromone value on that path is higher thereby increasing the pheromone quantity even further. This can be considered as special type of *reinforced learning* mechanism. Also, the chemical *pheromone* evaporates easily, thus reducing the pheromone level on both the paths. Ultimately the pheromone level on the longer path drops to nearly zero and hence all the ants use the smaller paths after some time. Note that the path from point B to food source is negligibly small and hence the pheromone level on it too increases rapidly and this becomes the selected path at point B instead of the longer path going back.

As mentioned earlier, Marco Dorigo modelled this behaviour of ants mathematically on small agents called *Artificial Ants*. These ants take probabilistic decisions over a set of possible solutions as a function of pheromone value associated with that path and heuristic information which is based on the input data. During each iteration, a solution is chosen with a probability which is given by (deCastro & Von Zuben, 2005):

$$\frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{h \in S} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}} \qquad (1)$$

In Equation (1), τ_{ij} refers to the pheromone level and η_{ij} is the heuristic information.

Equation (1) is called the *Random Proportional Transition Rule* (deCastro & Von Zuben, 2005). The pheromone and heuristic information can be weighted using α and β and hence, $\alpha, \beta \in R^+$. The pheromone value of a solution can also be updated on the basis of its performance by a process of *reinforcement*. The pheromone levels for good solutions can be reinforced by updating pheromone values on such solutions. This leads to faster convergence on optimal solutions.

In addition to this, pheromones are evaporated proportionally on all solutions. If ρ denotes the evaporation rate, the pheromone value of a solution is found by multiplying it by a factor of $(1 - \rho)$. The purpose of evaporation is opposite to that of reinforcement. It aims at avoiding too rapid convergence of the solution, which might lead to a sub-optimal solution (Dorigo & Stützle, 2003). Evaporation implements convenient *forgetting*, leading to exploration of a bigger solution space.

AntHocNet

AntHocNet (Di Caro, Ducatelle, & Gambardella, 2004) is an Ant Colony-based routing technique for ad-hoc networks. It is a hybrid algorithm which relies on reactive route set up combined with proactive route probing, maintenance and improvement. It is an on-demand routing algorithm, which means that the paths between the source and destination are set up as and where required and is not computed beforehand. *AntHocNet* is also a table-based routing protocol which means that the tables are used by each node to keep track of all the paths from that node to other nodes in the network. Every on-demand routing algorithm in ad-hoc networks has the following three functionalities (Toh, 2002).

1. Route Discovery: To find a route between the source and the destination.
2. Route Selection: If multiple routes are present, an algorithm selects the route(s) that are used for the purpose of routing. This might include selecting routes from a given table of routes.
3. Route Maintenance: To take suitable actions when routes break due to movement of nodes or link failure.

Though an algorithm may offer functions related to security, fault tolerance and other application-specific functionalities, these are the three basic functionalities present in almost every routing algorithm. We would now discuss how *AntHocNet* implements all the three functionalities.

Route Discovery

As mentioned earlier, an *AntHocNet* is an on-demand routing protocol. Routes are found 'on-the-go'. In *AntHocNet*, small control packets called *ant agents*

are used for transferring control information within the network. When a source node *s* wants to communicate with a destination node *d*, it looks for a path in its *pheromone table*. If it does not find any path, it broadcasts a *reactive forward ant* F_d^s. Ant agents, which are copies of each other (like the ones that are broadcasted), are said to be belonging to the same *generation* of ants. A node *i* that receives the ant, in turn, looks for a path to the destination *d* in its own pheromone table. A pheromone table stores entries in the form T_{nd}^i where T_{nd} represents the suitability of going over to neighbouring node *n* to reach *d*. If pheromone information is available, the next node is chosen on the basis of the *Random Transition Rule* given in Equation (1) (Di Caro, Ducatelle, & Gambardella, 2004). A node *n* is chosen with a probability P_{nd} (Di Caro, Ducatelle, & Gambardella, 2004).

$$P_{nd} = \frac{(T_{nd}^i)^{\beta_1}}{\sum_{j \in N_d^i}(T_{jd}^i)^{\beta_1}}, \quad \beta_1 \geq 1, \tag{2}$$

Here, N_d^i represents the set of neighbours over which path to the node *d* is known. On comparing Equation (1) with Equation (2), the reader would notice that the value of *α* denoting the importance of heuristic information is *zero* here and β_1 signifies the parameter which can be used to control the exploratory behaviour of ants.

If the node *i* does not have any entry in its pheromone table, it rebroadcasts the ant. To prevent flooding of the network by these broadcasts, when a node receives an ant which it had received before (that is, several ants of the same generation), it compares it with the ant with best performance. If the number of hops *h* and

the travel time $\ddot{T}_p$ is within the acceptance factor a_1, then only it is rebroadcast by that node. Otherwise, the ant is discarded. An exception to this rule is made for ants which differ in the first-hop with each other. For such ants, a higher acceptance factor a_2 is used. This prevents *kite-shaped paths (Pseudo multiple paths)* (Di Caro, Ducatelle, & Gambardella, 2004) as shown in Figures 3 and 4.

As an ant reaches the destination *d*, it is converted into a backward ant. A backward ant traverses the route *P* in the reverse direction, updating the value of pheromone at each node. The time taken to transport data packet over *P* is estimated as (Di Caro, Ducatelle, & Gambardella, 2004).

$$\hat{T}_p = \sum_{i=1}^{n-1} \hat{T}_{i+1}^i \tag{3}$$

Where $\hat{T}_{i+1}^i$ is the total time required to transfer Q (length of queue) + 1 packets to the MAC layer. This is found as (Di Caro, Ducatelle, & Gambardella, 2004)

$$\hat{T}_{i+1}^i = \left(Q_{mac}^i + 1\right)\hat{T}_{mac}^i \tag{4}$$

$\ddot{T}_{mac}^i$ is estimated as (Di Caro, Ducatelle, & Gambardella, 2004)

$$\ddot{T}_{mac}^i = \alpha\ddot{T}_{mac}^i + (1-\alpha)\hat{t}_{mac}^i \tag{5}$$

where $\alpha \in [0, 1]$.

If $\hat{T}_d^i$ is the travelling time estimated by an ant, pheromone value entry in table T^i at node *i*, given by

Figure 3. Pseudo multiple path (kite shaped path)[3]
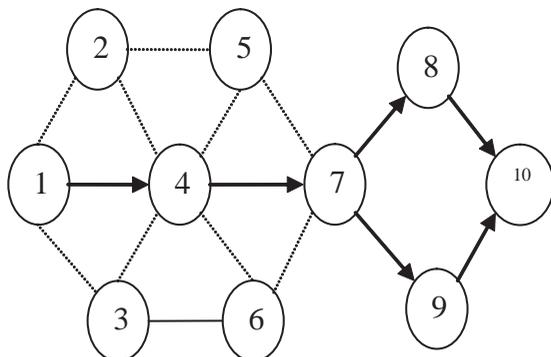
Figure 4. Paths with 1st hop different (accepted multiple paths)[4]
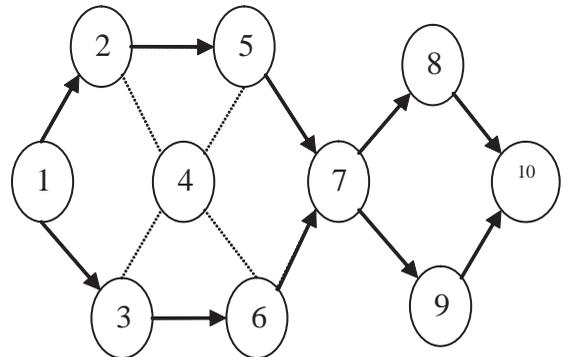
T_{nd}^i is updated as (Di Caro, Ducatelle, & Gambardella, 2004)

$$\tau_d^i = \left(\frac{\hat{T}_d^i + hT_{hop}}{2}\right)^{-1} \tag{6}$$

T_{hop} is a fixed value representing single hop time in unloaded conditions. The value of T_{nd}^i is updated as (Di Caro, Ducatelle, & Gambardella, 2004)

$$T_{nd}^i = \gamma T_{nd}^i + (1-\gamma)\tau_d^i \tag{7}$$

where $\gamma \in [0, 1]$.

Data Routing

Once the paths have been setup, routing is done *stochastically.* This means that when a node receives a data packet, it can choose to forward it to one of the several nodes in its list. A node chooses a next node from the table with a probability (Di Caro, Ducatelle, & Gambardella, 2004):

$$P_{nd} = \frac{(T_{nd}^i)^{\beta_2}}{\sum_{j\in N_d^i}(T_{jd}^i)^{\beta_2}} \tag{8}$$

If one compares Equation (8) with Equation (2), we find that, using *AntHocNet,* data packets are forwarded in a manner similar to control packets (reactive forward ant packets in this case). However, the value of β_2 is chosen much higher than β_1 so as to give preference to better paths while routing the data. On the other hand, by lowering the value of β_1 we can give liberty to reactive forward ants to be more explorative in nature.

Even though a node prefers to forward packets to a next node which lies on a better path, as the load on a network rises, the pheromone value on this particular path starts decreasing and hence the probability of selecting other paths increases. This kind of stochastic data routing leads to *automatic load balancing*.

Route Maintenance

As mentioned earlier, *AntHocNet* uses proactive route probing and maintenance. Source node unicasts proactive ants to the destination. These ants are forwarded by each node in a manner similar to the other control packets. However, they also have a small probability

of being broadcasted. Hence, an ant which reaches the destination node after selected number (say, one or two) of broadcasts finds out a fresh route to destination. If an ant is not broadcasted, it finds out fresh information regarding this route.

A node uses *hello* packets to track its neighbours. If a node finds out a new neighbour, it adds it to its routing table. However, if a node discovers that a node has moved from its neighbourhood (for example, if it does not respond to the *hello* messages), a node broadcasts its neighbour, it removes it from its routing table and broadcasts a *link failure notification*, so that the other nodes in the network update their table to include that the node does not have a route to the destination any more.

If, however, the node realizes that the problem is due to link failure and not node movement, it tries to repair the path by initiating a *route repair ant* to the destination. This ant tries to find an alternate path to the destination and if it does not return within a specified time, the node assumes a failure, and broadcasts *link failure notification.*

CONCLUSION

This chapter presents an example of how ACO techniques can be applied to solve problems in ad-hoc networks. We first introduced the food foraging behaviour of ants and explain the phenomenon using the concept of pheromone. We then explained the concept of artificial ants and other ACO techniques and explain the *random probability rule*.

We, then, introduce the *AntHocNet* algorithm (Di Caro, Ducatelle, & Gambardella, 2004) that uses ACO techniques, for routing in mobile ad-hoc networks. We explained how different routes are discovered using *reactive forward ants*. We also explained how *random probability rule* is used in the case of *AntHocNet*. We then presented how data can be routed stochastically using these routes and the *automatic load-balancing* that it achieves. In the end, we explained how these routes are maintained using *hello packets* and *link failure notification*.

1534

REFERENCES

Abdul-Razaq, T. S., Potts, C. N., & Van Wassenhove, L. N. (1990). A survey of algorithms for the single machine total weighted tardiness scheduling problem. *Discrete Appied Mathematics.* 26, 2-3 (Feb. 1990) (pp. 235-253). Elsevier Science Publishers B. V. Amsterdam, The Netherlands.

Ai-bin, C., Zi-xing, C., & De-wen, H. (1993). Clustering in mobile ad hoc network based on neural network. *Journal of Central South University of Technology* , 699-702.

Barolli, L., Koyama, A., & Shiratori, N. (2003). A QoS Routing Method for Ad-Hoc Networks Based on Genetic Algorithm. *14th International Workshop on Database and Expert Systems Applications(DEXA)* (pp. 175-179). Prague: IEEE Computer Society.

Cormen, T. H., Leiserson, C., E. & Rivest, R. L. (2001). Introduction to Algorithm, Second Edition, MIT Press, ISBN 0-262-03293-7, Chapter 30.

de Castro, L. N., & Von Zuben, F. J. (2005). *Recent Developments in Biologically Inspired Computing.* Idea Group Publishing.

Deneubourg, J.-L., S., A., S., G., & J., P. (1990). The self-organizing exploratory pattern of the Argentine Ants. *Journal of Insect Behavior* , 159–168.

Di Caro, G., Ducatelle, F., & Gambardella, L. M. (2004). AntHocNet: An Adaptive Nature-Inspired Algorithm. *Lecture Notes In Computer Science* , 461-470.

Dorigo, M., & Stützle, T. (2003). The Ant Colony Optimization Metaheuristic: Algorithm, Applications and Advances. In F. Glover, & K. G. A., *Handbook of Metaheuristics* (pp. 251-285). Kluwer Academic Publishers.

Dressler, F. (2006). *Self-Organization in Ad Hoc Networks: Overview and Classification.* University of Erlangen, Dept. of Computer Science.

Goss, S., Aron, S., Deneubourg, J.-L., & Pasteels, J. (1989). Self-organized shortcuts in the argentine ant. *Naturwissenschaften* , 579-581.

Murthy, C. S., & S., M. B. (2004). *Ad Hoc Wireless Networks.* Upper Saddle River, NJ: Prentice Hall.

Toh, C. K. (2002). *Ad Hoc Mobile Wireless Systems.* Prentice Hall PTR.

Toth, P. and Vigo, D. (2001). An overview of vehicle routing problems. In *The Vehicle Routing Problem*, P. Toth and D. Vigo, Society for Industrial and Applied Mathematics (pp. 1-26). Philadelphia, PA,

Vesel, A., & Zerovnik, J. (2000). How good can ants color graphs? *Journal of Computing and Information*, 131-136.

Vicente, E., Mujica, V., Sisalem, D., & Popescu-Zeletin, R. (2005). NEURAL: A Self-organizing Routing Algorithm for Ad Hoc networks. *Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks* (pp. 259-266). Trentino: IEEE Computer Society.

Wedde, H., & Farooq, M. (2005). The wisdom of the hive applied to mobile ad-hoc networks. *IEEE Swarm Intelligence Symposium*, (pp. 341 - 348). Pasadena.

Zhang, Y., & Lee, W. (2000). Intrusion detection in wireless ad-hoc networks. *International Conference on Mobile Computing and Networking*, (pp. 275 - 283). Boston.

KEY TERMS

Ant Colony Optimization: Ant Colony Optimization involves a set of algorithms modelled on the foraging behaviour of a colony of natural ants.

AntHocNet: An Ant Colony Optimization-based algorithm for routing in ad-hoc networks using reactive route set up combined with proactive route probing, maintenance and improvement.

Heuristic Information: Static value associated with a solution that represents the relative suitability of a solution among its peers based on intuition, previous experience or common sense.

Pheromone: Chemical secreted by natural ants, the presence of which is an indicative of the number of ants that have followed a particular path. This chemical is modelled to represent the historical preference that is associated with a path in Ant Colony Optimization.

Proactive Forward Ant: Control ant agents that are unicast to destination node and are responsible for finding fresh information about existing routes or to find fresh nodes to the destinations.

Reactive Forward Ant: Ant agents responsible for discovering paths to the destination nodes.

Stigmergy: Method of indirect communication between simple agents by altering their environment. Ants use a chemical called Pheromone to communicate with each other, which is an example of stigmergy.

Swarm Intelligence: Group of bio-inspired algorithms which is modelled on the collective behaviour of a group of social organisms such as ants, termites, bees, fishes and birds.

ENDNOTES

[1&2]   Based on Di Caro, Ducatelle, & Gambardella, 2004

[3&4]   Based on Di Caro, Ducatelle, & Gambardella, 2004

Swarm Robotics

Amanda J.C. Sharkey
University of Sheffield, UK

INTRODUCTION

Swarm Robotics is a biologically inspired approach to the organisation and control of groups of robots. Its biological inspiration is mainly drawn from social insects, but also from herding and flocking phenomena in mammals and fish. The promise of emulating some of the efficient organisational principles of biological swarms is an alluring one. In biological systems such as colonies of ants, sophisticated cooperative behaviour emerges despite the simplicity of the individual members, and the absence of centralised control and explicit directions. Such societies are able to maintain themselves as a collective, and to accomplish coordinated actions such as those required to construct and maintain nests, to find food, and to raise their young. The central idea behind swarm robotics is to find similar ways of coordinating and controlling collections of robots.

BACKGROUND

The mechanisms that underlie social insect behaviour have inspired an approach that emphasises autonomy, emergence and distributed functioning, and avoids a reliance on centralised control and communication. This approach underlies both swarm robotics, and the closely related notion of artificial "swarm intelligence". The term "swarm intelligence" was first coined in the context of cellular robotic systems, on the basis of the features that the simulated robotic collections shared with social insects: namely "decentralised control, lack of synchronicity, simple and (quasi) identical members" and size (Beni and Wang, 1989). Bonabeau et al (1999) describe as swarm intelligence, "any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies" (pg 7, Bonabeau et al, 1999). The key ingredients of swarm intelligence that they emphasise are self-organisation, and stigmergy,

(indirect communication via the environment). Martinoli (2001) similarly describes the swarm intelligence approach as emphasising "parallelism, distributedness, and exploitation of direct (agent-to-agent) or indirect (via the environment) local interactions among relatively simple agents.

Swarm robotics has been described as the application of swarm intelligent principles to collective robotics (Sharkey and Sharkey 2006). The same principles of decentralised local control and communication are applied to physically instantiated robots. In swarm robotics, the emphasis is on using a number of simple robots that are autonomous, not subject to global control, and that have limited communication abilities. The reliance on local communication means that the potential problems of communication bottlenecks, or centralised failure, are avoided. The system benefits from the redundancy of using several robots: if individual robots were to fail, others could take over, and new ones could be added without the need for recalibration of communicative systems. In the same way, the activities of an ant colony need not be affected by the removal of some of its members. The simplicity of the individual robots means that they are able to respond quickly to the environment. There are also several tasks, such as exploring an environment, that can be accomplished more efficiently if a number of robots are used.

Of course, using a collection of robots creates some new problems itself (Bonabeau et al, 1999). There is the possibility of stagnation: without global knowledge, a group of robots can find themselves in a deadlock situation. Too many robots trying to reach the same location, or perform the same task could obstruct each other. Another problem is finding a solution to a task: how can situations be engineered in order that a desired solution can emerge? Nonetheless, the promise of being able to send a number of autonomous robots to perform a task, particularly in sites that are remote and inhospitable to humans, outweighs the disadvantages.

SWARM ROBOTICS

Early work in swarm robotics can be illustrated by describing a series of studies in which simple robots are shown to be able to collect a number of objects in one place, and even to sort them. This work was initiated by a paper by Deneubourg et al (1991), and observations of the ability of ants to work together to sort their brood into clusters of eggs, larvae and cocoons, despite the insects' limited communicative abilities. In their simulations, "ant-like robots" (ALRs) moved randomly in a two dimensional environment populated by objects, and showed a greater probability of picking up the isolated items they encountered, and a greater probability of dropping them at locations where more items of that type are present. Their simulations demonstrated that the model eventually resulted in clustering and sorting of objects. Beckers et al (1994) applied these ideas to actual robots. Their robots had IR sensors for obstacle avoidance, a gripper to pick up the objects, and a microswitch that was activated when they pushed three pucks or more. They could (i) travel in a straight line until (ii) an obstacle was detected, whereupon they would turn to avoid it, or (iii) until their micro switch was activated, whereupon they would drop the pucks they were carrying, and turn away. Since the robots` grippers would automatically collect up pucks they encountered, these behaviours were sufficient to result in the eventual collection of all the objects in a single cluster. Holland and Melhuish (1999) extended these results: augmenting the robots` behaviours with a "pull-back" rule that required robots to pull pucks of one colour back for some distance before releasing them. Its effect was that (after several hours), pucks scattered across the arena were collected up by the robots, and sorted into clusters of different colours. More recently, Wilson et al (2004) reported further investigations of different minimalist solutions to 'ant-like annular sorting' using simple robots and simple mechanisms.

Other swarm robotic studies have also explored the behaviours that can be accomplished by robots that respond in a fixed manner to environmental stimuli, and that do not directly communicate with each other. A number of studies were designed to investigate explicitly cooperative tasks, (tasks that have been designed to require cooperation), such as pushing a box that is too heavy to be pushed by a single robot (Kube and Zhang, 1996; Kube and Bonabeau, 2000). Stick pulling (Ijspeert et al, 2001) is a similarly explicitly cooperative task that involved locating sticks in a circular arena and pulling them out of the ground in circumstances where the length of the stick means that a single robot cannot pull it out by itself, but must collaborate with a second robot. Ijspeert et al (2001) used reactive robots with minimal sensing abilities. Their results show that collaboration can still be obtained despite the absence of signalling, planning, or direct communication.

These studies share a number of features. They all involve a number of robots. The robots are autonomous, and not controlled centrally; the control methods used could be scaled up to larger numbers of robots, or scaled down to smaller numbers since each robot performs a set number of fixed behaviours in response to certain stimuli. The individual robots are certainly simple – they have no knowledge of the environment they are in, or even of the other robots in it. They are essentially reactive: they have no knowledge or map of their environment, and they have no ability to communicate directly with other robots, or to receive instructions. Nonetheless, they exhibit apparently cooperative behaviour. Many of the studies make use of the concept of stigmergy, a term introduced by Grassé (1959) in the context of his observations of termite building behaviour. He noted that termite workers were stimulated to further constructive activity in the presence of particular features of a construction. The behaviour of the termite is affected by changes in the environment created either by itself, or by other termites: a form of indirect communication, where environmental changes have a signalling function. All of the examples discussed here explicitly draw analogies and parallels to living biological systems. Together, they illustrate some of the potential of swarm robotics: despite the simplicity of the individual robots, their interactions with the environment result in the performance of tasks in the physical world, and demonstrate that cooperation between such simple entities can emerge in the absence of any planning, centralised coordination, or even any direct communication between the robots.

Nonetheless, as research in swarm robotics has developed, so has a certain lack of clarity and agreement about the terms to be used and about what their defining features are (see also Dorigo and Sahin, 2004). There is agreement that swarm robotics implies the use of control and communication methods that are decentralised and scalable, so that communication bottlenecks are avoided, the robots operate autonomously, and the

same approach could be applied unchanged to varying sizes of robot collection. It is less clear whether swarm robotics necessarily implies the use of reactive control and constraints on the kinds of communication involved. Bonabeau et al (1999) suggest that swarm-based robotics may be "loosely defined" as "reactive collective robotics" (pg 19), and that "swarm-based robotics relies on the anti-classical AI idea that a group of robots may be able to perform tasks without explicit representations of the environment and of the other robots". Should swarm robotics be restricted to the use of robots with such minimal representational abilities?

Arguments in favour of restricting swarm robotics to the use of reactive robots can be made on the basis of parsimony. Pfeifer and Scheier (2001) argue that a more parsimonious model should be preferred over more complicated ones, and that, for instance, a model that can explain "*the clustering behaviour of ants based on simple reflexes, for example, is to be preferred over one that postulates some sort of internal representation of clustering*". Of course, there are practical advantages to be gained from attempting to accomplish a given task with the simplest possible mechanism. Minimalist unit design and a reliance on reactive robots that respond rapidly to stimuli in the environment can facilitate a rapid response to changing situations, and lead to the use of robots that are relatively cheap and expendable. Wilson et al (2004) for example, provide a practical justification for their minimalist approach that relies on robots built from simple mechanical components and sensors, claiming, "*Potentially, this allows for the production of more robust and cheaper robot units…. Simple behavioural rules are employed so the robots are less complex. Rules have to be embodied and realized in a machine, and the more complicated the rules, the more complicated the hardware and software in the machine is likely to be. The more complicated the hardware and software required, the more there is to go wrong*." The problem is that there is likely to be a limit to the number and kinds of task that can be accomplished using reflexive behaviours, and avoiding internal representation.

Another reason for an emphasis on reactive robots in swarm robotics is based on its inheritance from behaviour-based robotics. Rodney Brooks and his associates introduced the idea of behaviour-based robotics, and the advantages to be gained from departing from the traditional emphasis on reasoning and representation in Artificial Intelligence (Brooks 1999). They showed that certain tasks could be solved more easily by robots that were situated in the world, and could react to and exploit characteristics of the environment, than by robots such as Shakey (Nilsson, 1984) that depended on a representationally intensive, and slow approach to modelling the world. However, again the range of tasks to which robots without representations can be applied is limited, and more recent formulations of behaviour-based robotics (Mataric 1997) incorporate the idea of action-centred representations.

A final reason for preferring to use reactive robots in swarm robotics can be found in assumptions about the limited abilities of the social insects that inspire them. Debates over parsimonious explanation have always occurred in biology. For instance, Griffin (1992) has argued that there has been a long held view that social insects are little more than "genetically programmed clockwork". Under such a view, insects can only react to stimuli, and are not able to represent the world, or to communicate amongst themselves. This view of insects as clockwork (albeit clockwork with sensors that enable it to respond to the world) is one that can be traced back to behaviourism's response to the anthropomorphism that preceded it. The synopsis of a book on insect learning (Papaj and Lewis, 1992) claims that "until recently, insects were viewed as rigidly programmed automatons: now however, it is recognised that they can actually learn and that their behaviour is plastic".

There is a gradually accumulating body of evidence that shows that social insects do have some representational and learning abilities, and that their communicative abilities are more extensive than was once supposed. For example, Collett and Collett (2002) review evidence for memory use in insect visual navigation, describing evidence that shows reliable recognition of visual landmarks, and reliable performance of learned routes. Franks and Richardson (2006) report evidence that the ant *Temnothorax albipennis* can use tandem running to lead another ant from nest to food, and make use of bi-directional feedback, as the leader ant modifies its behaviour when being followed – the leader teaches the route to the follower. Robinson et al (2005) have shown that as well as laying a pheromone trail to guide others to a food sources, Pharaoh's ants can also lay a negative "no entry" signal to mark an unrewarding trail path.

Such findings could justify extending the capabilities of individual robots in a swarm beyond those of

reactive control. Robots with some ability to learn a route, to recognise landmarks, or to keep track of the number of encounters they have with others would be able to perform a wider range of tasks. It would be interesting to explore the ways in which such abilities could be incorporated into swarm robotics. An approach of "biologically plausible minimalism", in which the representational and communicative abilities of the robots were restricted to those plausible for social insects would ensure that any such approach still preserved the swarm advantages of decentralisation and scalability shown in their biological counterparts.

FUTURE TRENDS

An avenue that could be explored in future swarm robotic research is that of incorporating simple forms of memory and representational ability, without compromising the swarm-related benefits of local control and communication and scalability. Relatively simple robots could, for instance, be given some minimal representational abilities: the ability for instance to learn a route, or to recognise landmarks. Similarly, it would be interesting to explore the use of some further communicative abilities other than that of pheromone trail laying. For example, robots could be given the ability to convey, and to sense, the tasks that they and other robots are involved in, and to keep account of the frequency of their encounters. This would enable some distributed decision making abilities, and dynamic switching between tasks based on their local records of the numbers performing each task. These limited cognitive abilities would still depend on entirely local control, and would be scalable, but such extensions could be used to extend the range and complexity of tasks to which swarm robotics could be applied.

CONCLUSION

In this article, we have surveyed swarm robotics research and discussed the source of its biological inspiration – the self-organised behaviour of social insects. Some representative studies have been described, and their common characteristics noted. These include the ideas that the robots in a swarm should be simple, autonomous, and subject to local control and communication. The expected benefits of using such robots are that they

should be able to provide a robust and flexible solution for practical applications in inaccessible areas; one that benefits from an inherent redundancy, since robots could fail or be replaced without the need for recalibration of the control and communication methods.

The approach is of interest, but still in its early stages. There is still some disagreement about the use of the term 'swarm robotics', and the constraints it implies. In particular, it is not clear whether swarm robotics necessarily involves the use of reactive robots with effectively no representational ability. There are reasons to prefer the simplest possible solution for a given task, but the argument is made here that there is evidence that social insects do have some ability to represent the environment, and that incorporating such abilities into swarm robotics would extend the range of tasks to which the approach could be applied, without compromising its swarm-related advantages.

REFERENCES

Beckers, R., Holland, O.E. and Deneubourg, J.L. (1994) From local actions to global tasks: Stigmergy and collective robotics. In *Proceedings A-Life IV* MIT Press

Beni, G., and Wang, J. (1989) Swarm intelligence. In *Proceedings. of the Seventh Annual Meeting of the Robotics Society of Japan*, Tokyo, Japan, p 425-428.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999) *Swarm Intelligence: From Natural to Artificial Systems,* Oxford University Press.

Brooks, R. (1999) *Cambrian Intelligence: The Early History of the New AI* A Bradford Book: MIT Press.

Collett, T.S. and Collett, M.(2002) Memory use in insect visual navigation. Nature Reviews. *Neuroscience*, 3, 542-552.

Deneubourg, J.-L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. and Chretien, L. (1991) The dynamics of collective sorting: Robot-like ants and ant-like robots. In J.A. Meyer and S.W. Wilson (Eds), *From animals to animats: Proceedings of the First International Conference on Simulation of Adaptive Behaviour* (pp 356-363, Cambridge, MA: MIT Press (A Bradford Book).

Dorigo, M. and Sahin, E. (2004) Guest editorial: Swarm robotics. *Autonomous Robots*, 17, 2-3, 111-113.

Franks, N.R. and Sendova-Franks, A.B. (1992) Brood sorting by ants: Distributing the workload over the work-surface. *Behavioural Ecology and Sociobiology*, 30, 109-123.

Franks, N.R.and Richardson, T. (2006) Teaching in tandem-running ants. *Nature*, 439, 153

Grey Walter, W. (1954) The Living Brain. A Pelican Book.

Grassé, P.P. (1959) La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes Natalensis et Cubitermes sp La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeur. *Insectes Sociaux*, 6, 41-80.

Griffin, D.R. (1992) *Animal Minds.* The University of Chicago Press, Chicago and London.

Holland, O, and Melhuish, C. (1999) Stigmergy, Self-Organisation, and Sorting in Collective Robotics. *Artificial Life* 5, 173-202.

Ijspeert, A.J., Martinoli, A., Billard, A., Gambardella, L.M. (2001) Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots,* 11, 149-171.

Kube, C. and Zhang, H. (1996) The use of perceptual cues in multi-robot box-pushing. In *Proceedings of IEEE International Conference on Robotics and Automation,* pp2085-2090.

Kube, C.R. and Bonabeau, E. (2000) Cooperative transport by ants and robots. *Robotics and Autonomous Systems* 30, 85-101

Martinoli, A. (2001) Collective complexity out of individual simplicity. Invited book review on "Swarm Intelligence: From Natural to Artificial Systems" by E.Bonabeau, M. Dorigo, and G. Theraulaz, *Artificial Life,* 7, 3, pp 315-319.

Mataric, M. (1997) Behaviour-based control: examples from navigation, learning and group behaviour. *Journal of Experimental and Theoretical AI, Special Issue on Software Architectures for Physical Agents,* 9, 2-3, (Eds) H. Hexmoor, I. Horswill, and D. Kortenkamp, 323-336.

Nilsson, N. (1984) Shakey the robot. *Technical Report Technical Note 323*. Menlo Park, CA: SRI International

Papaj, D.R., and Lewis, A. C., (1992) (Eds) *Insect Learning: Ecological and Evolutionary Perspectives*, New York: London, Chapman and Hall.

Pfeifer, R. and Scheier, C. (2001) *Understanding Intelligence*. MIT Press, London, England.

Robinson, E., J.H, Jackson, D.E., Holcombe, M., Ratnieks, F.L.W. (2005) "No entry" signal in ant foraging. *Nature*, 438, 442-442 Brief Communications

Sharkey, A.J.C. and Sharkey, N.E. (2006) The application of swarm intelligence to collective robotics. In J. Fulcher (Ed) *Advances in Applied Artificial Intelligence*, Hershey, PA: Information Science Publishing, 157-185.

Wilson, M., Melhuish, C., Sendova-Franks, A.B., and Scholes, S. (2004) Algorithms for Building Annular Structures with Minimalist Robots Inspired by Brood Sorting in Ant Colonies. *Autonomous Robots*, 17, 115-136.

KEY TERMS

Behaviour-Based Robotics (BBR): A paradigm initiated by Brooks (1999) that stressed the importance of studying robots situated in the world, and responding to information directly gathered by their sensors. BBR robots make minimal use of internal representations.

Emergent Behaviour: Results from the unsupervised interaction of a number of simpler processes. The complex behaviour of an ant colony is a good example of emergent behaviour: the individual ants carry out their tasks on a local level, but the combined effect is of a colony that is able to maintain its own organisation.

Reactive Robotics: An approach to robot control in which there is a direct mapping from the sensor input to the robot, and its motor output. No use is made of internal representations of the world. The approach dates from the reactive robots developed by Grey Walter (1954).

Self-Organisation: Pattern-formation processes in physical and biological systems that occur as a result of

interactions internal to the system, without intervention by external influences.

Stigmergy: A method of indirect communication that occurs when one individual modifies the environment, and another responds to that environment at a later time.

Swarm Intelligence: Describes attempts to design algorithms and to solve problems, using methods inspired by observations of the collective behaviour of biological groups such as insect colonies.

Social Insects: Insects that live cooperatively in colonies and exhibit a division of labour among distinct castes. E.g. termites, ants, bees, some wasps.

Symbol Grounding Problem

Angelo Loula
State University of Feira de Santana, Brazil
State University of Campinas (UNICAMP), Brazil

João Queiroz
State University of Campinas (UNICAMP), Brazil
Federal University of Bahia, Brazil

INTRODUCTION

The topic of representation acquisition, manipulation and use has been a major trend in Artificial Intelligence since its beginning and persists as an important matter in current research. Particularly, due to initial focus on development of symbolic systems, this topic is usually related to research in symbol grounding by artificial intelligent systems. Symbolic systems, as proposed by Newell & Simon (1976), are characterized as a high-level cognition system in which symbols are seen as "[lying] at the root of intelligent action" (Newell and Simon, 1976, p.83). Moreover, they stated the Physical Symbol Systems Hypothesis (PSSH), making the strong claim that "a physical symbol system has the necessary and sufficient means for general intelligent action" (p.87).

This hypothesis, therefore, sets equivalence between symbol systems and intelligent action, in such a way that every intelligent action would be originated in a symbol system and every symbol system is capable of intelligent action. The symbol system described by Newell and Simon (1976) is seen as a computer program capable of manipulating entities called symbols, 'physical patterns' combined in expressions, which can be created, modified or destroyed by syntactic processes. Two main capabilities of symbol systems were said to provide the system with the properties of closure and completeness, and so the system itself could be built upon symbols alone (Newell & Simon, 1976). These capabilities were designation – expressions designate objects – and interpretation – expressions could be processed by the system. The question was, and much of the criticism about symbol systems came from it, how these systems, built upon and manipulating just symbols, could designate something outside its domain.

Symbol systems lack 'intentionality', stated John Searle (1980), in an important essay in which he de-

scribed a widely known mental experiment (*Gedankenexperiment*), the 'Chinese Room Argument'. In this experiment, Searle places himself in a room where he is given correlation rules that permits him to determine answers in Chinese to question also in Chinese given to him, although Searle as the interpreter knows no Chinese. To an outside observer (who understands Chinese), the man in this room understands Chinese quite well, even though he is actually manipulating non-interpreted symbols using formal rules. For an outside observer the symbols in the questions and answers do represent something, but for the man in the room the symbols lack intentionality. The man in the room acts like a symbol system, which relies only in symbolic structures manipulation by formal rules. For such systems, the manipulated tokens are not about anything, and so they cannot even be regarded as representations. The only intentionality that can be attributed to these symbols belongs to who ever uses the system, sending inputs that represent something to them and interpreting the output that comes out of the system. (Searle, 1980)

Therefore, intentionality is the important feature missing in symbol systems. The concept of intentionality is of *aboutness,* a "feature of certain mental states by which they are directed at or about objects and states of affairs in the world" (Searle, 1980), as a thought being about a certain place.[1] Searle (1980) points out that a 'program' itself can not achieve intentionality, because programs involve formal relations and intentionality depends on causal relations. Along these lines, Searle leaves a possibility to overcome the limitations of mere programs: 'machines' – physical systems causally connected to the world and having 'causal internal powers' – could reproduce the necessary causality, an approach in the same direction of situated and embodied cognitive science and robotics. It is important to notice that these 'machines' should not be just robots controlled

by a symbol system as described before. If the input does not come from a keyboard and output goes to a monitor, but rather came in from a video camera and then out to motors, it would not make a difference since the symbol system is not aware of this change. And still in this case, the robot would not have intentional states (Searle 1980).

Symbol systems should not depend on formal rules only, if symbols are to represent something to the system. This issue brought in another question, how symbols could be connected to what they represent, or, as stated by Harnad (1990) defining the Symbol Grounding Problem:

"How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads? How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?"

The Symbol Grounding Problem, therefore, reinforces two important matters. First that symbols do not represent anything to a system, at least not what they were said to 'designate'. Only someone operating the system could recognize those symbols as referring to entities outside the system. Second, the symbol system cannot hold its closure in relating symbols only with other symbols; something else should be necessary to establish a connection between symbols and what they represent. An analogy made by Harnad (1990) is with someone who knows no Chinese but tries to learn Chinese from a Chinese/Chinese dictionary. Since terms are defined by using other terms and none of them is known before, the person is kept in a 'dictionary-go-round' without ever understanding those symbols.

The great challenge for Artificial Intelligence researchers then is to connect symbols to what they represent, and also to identify the consequences that the implementation of such connection would make to a symbol system, e.g. much of the descriptions of symbols by means of other symbols would be unnecessary when descriptions through grounding are available. It is important to notice that the grounding process is not just about giving sensors to an artificial system so it would be able to 'see' the world, since it 'trivializes' the symbol grounding problem and ignores the important

issue about how the connection between symbols and objects are established (Harnad, 1990).

BACKGROUND

The symbol grounding problem aroused from the notice that symbol systems manipulated structures that could be associated with things in the world by an observer operating the system, but not by the system itself. The quest for symbol grounding processes is concerned with understanding processes which could enable the connection of these purely symbolic representations with what they represent in fact, which could be directly, or by means of other grounded representations.

This represents a technological challenge as much as a philosophical and scientific one, but there is a strong interrelation between them. From one side there is the concern with the technological design and engineering of symbol grounding processes in artificial systems. On the other side, the grounding process is a process present in natural systems and therefore precedes artificial systems. Theories and models are developed to explain grounding and if consistent and detailed enough may in principle be implemented in artificial systems, which in return correspond to a laboratory for these theories, when their hypothesis are tested and new questions are raised, allowing further refinement and experimentation.

A first proposal for symbol grounding as made by Harnad (1990) in the same paper where he gave a definition for the 'symbol grounding process'. Harnad proposed that symbolic representations should be grounded bottom-up by means of non-symbolic representations: iconic representations – sensory projections of objects – and categorical representations – invariant features of objects. Neural networks were pointed out as a feature learner and discriminator, which could link sensory data with symbolic representations, after been trained to identify the invariant features. This would causally connect symbols and sensory data, but this proposal describes just a tagging system that gives names to sensed objects but does not use this to take actions and interact with its environment. A 'mental theater' is formed as Dennett (1991) defined, where images are projected internally and associated with symbols, but no one is watching it. Besides the symbols and the iconic representations are probably given by the systems operator and the system must learn them

all, making no distinction between them and attributing no functionality to them.

Another approach to deal with the limitation of symbol systems was presented by Brooks (1990). Instead of modeling artificial systems as symbol systems, Brooks rejected the symbolic approach for cognition modeling and the need of representations for this end: "[r]epresentation is the wrong unit of abstraction in building the bulkiest parts of intelligent systems"(Brooks, 1991b, p.139), with representations seen as centralized, explicit and pre-defined structures. He proposed the Physical Grounding Hypothesis (Brooks, 1990), supporting that intelligent systems should be embedded in the real world, sensing and acting in it, establishing causal relations with perceptions and actions, being built in a bottom up manner with higher levels depending on lower ones. There was no need for representations because the system is already in touch with the objects and events it would need to represent. Moreover, Brooks called up attention that the most important aspect of intelligence was left out: to deal with the world and its dynamics. Instead of dealing with sophisticated high-level processes dealing with simple domains, research in the so-called Nouvelle AI should focus in simpler processes dealing with greatly complicated domains (such as the real world) and work its way from there to higher level ones (Brooks, 1990). Brooks (1991a) also stated principles to this new approach, such as situatedness and embodiment, which are the mottos of the situated and embodied cognition studies (Clark, 1997).

Symbolic representations are in fact not incompatible with the Physical Grounding Hypothesis and with the Situated and Embodied Cognition approach. Brooks (1990) himself pointed out high-level abstractions should be made 'concrete' by means lower-level processes, thus symbolic representations should be causally constructed from the situated/embodied interaction dynamics through the artificial agent's history. This approach was followed by several researchers dealing with symbol grounding when building artificial systems in which representations emerge from agent's interactions, when learning processes take place (e.g. Ziemke 1999, Vogt 2002, Cangelosi 2002, Roy 2005; see also Christiansen & Kirby 2003; Wagner, 2003, for a review of experiments about language emergence).

In most of these new systems, artificial agents are situated in an environment, in which they can sense and act, and are allowed to interact with other agents

– either artificial or biological ones. By means of associative learning mechanisms, agents are able to gradually establish relations between representations and what they represent in the world, using communication as the basis to guide this learning process. And remarkably, when explicitly discussing the 'symbol grounding problem' (Vogt 2002, Cangelosi et al. 2002, Roy 2005), the sign theory of Charles Sanders Peirce, particularly his definition of a symbol, is brought forth as the theoretical background for a new view into this problem.

SEMIOTICS AND SYMBOL GROUNDING PROBLEM

The symbol grounding problem is fundamentally a matter of how certain things can represent other things to someone. Although symbol systems were said to have 'designation' properties (Newell & Simon, 1976), which would allow symbols manipulated by the system to stand for objects and events in the world, this property should actually be attributed to an outside observer who was the only one able to make this connection. The artificial system itself did not have this capability, so the symbols it manipulated were said to be ungrounded. Building artificial systems based on the hypothesis that symbolic processes were autonomous and no other process was required proved to be flawed, and the quest to understand representation processes came up as a major issue.

Representation is the focus of semiotics, the 'formal science of signs' as defined by Charles Sanders Peirce. His definition of Semiotics and his pragmatic notion of meaning as the 'action of signs' (semiosis), have had deep impact in philosophy, psychology, theoretical biology, and cognitive sciences. Sign model and classification was developed by Peirce from his logical-phenomenological categories. His definition of a sign as "something which stands to somebody for something in some respect or capacity" (Peirce 1931-1958, §2.228) interrelates three distinct elements: a sign, an object, which the sign represents in some respect, and an effect (interpretant) on an interpreter. The nature of the relation between sign and object establishes the 'most fundamental division of signs': signs can either be icons, indexes or symbols. Icons stand for the object through resemblance or similarity, since it carries properties in common with the object. The drawing

of an object, a diagram and 'sensory projections' are regarded as icons. Indexes establish spatio-temporal physical relation with its object, both occur as events and the interpreter is not responsible for connection between them, he just remarks it when it is established (Peirce 1931-1958, §2.299). Examples of indexes are smoke, which is related with fire, a scream that calls up our attention, or a bullet hole. According to Peirce, a symbol is a sign because it is interpreted as such, due to a natural or conventional disposition, in spite of the origin of this general interpretation rule (Peirce 1931-1958, §2.307). A word, a text and even a red light in a traffic light alerting drivers to stop are symbols. In this symbolic process, the object which is communicated to the interpretant through the sign is a lawful relationship between a given type of sign and a given type of object. Generally speaking, a symbol communicates a law to the interpretant as a result of a regularity in the relationship between sign and object.

Furthermore, it is important to remark that symbols, indexes and icons are not mutually exclusively classes; they are interrelated and interdependent classes. "A Symbol is a law, or regularity of the indefinite future. […] But a law necessarily governs, or "is embodied in" individuals, and prescribes some of their qualities. Consequently, a constituent of a Symbol may be an Index, and a constituent may be an Icon" (Peirce 1931-1958, §2. 293). Symbols require indexes which require icons. Harnad (1990) already noticed that symbols need non-symbolic representations and proposed that symbols are to be connected with sensory projections and categorical features, both regarded as icons in Peirce's theory. A symbol is a sign and as such it involves an object which it refers to and an interpretant, the effect of the sign, so a symbol can only represent something to someone and when someone is interpreting it. A symbol distinguishes itself from other signs since it holds no resemblance or spatial-temporal relation with the object and thus depends on a general rule or disposition from the interpreter. At last, symbols incorporate indexes (and icons, consequently) and one way symbols can be acquired is by exploiting indexical relations between signs and objects, establishing regularities between them.

FUTURE TRENDS

The discussion around the symbol grounding problem has an important component of theoretical aspects since it involves issues such as representation and cognitive modeling. Nevertheless, it is also a technological concern if researchers in Artificial Intelligence intend to model and build artificial systems which are capable of handling symbols in the appropriate way. The most evident consequence of discussion the problem of symbol grounding and ways of solving it is related to language and more generally with communication systems between agents (artificial ones or not). If we expect a robot to act appropriately when we say 'bring me that cup', we should expect it to know what these symbols represent so it will act accordingly. Moreover, we expect an artificial agent to learn and establish symbol-object connections autonomously, without the need of programming everything prior to the robot execution or reprogram it every time a new symbol is to be learned.

The employment of strong theoretical basis that describes thoroughly the process of interest can certainly contribute to the endeavor of modeling and implementing it in artificial systems. The semiotics of Charles S. Peirce is recognized as a strongly consistent theory, and has been brought forth by diverse researchers in Artificial Intelligence, though fragmentally. We expect that Peirce description of sign processes will shed light on the intricate problem of symbol grounding. Particularly, Peirce conception of meaning as sign action can open up perspectives on the implementation of semiotic machines, which can produces, transmits, receives, computes, and interprets signs of different kinds, meaningfully (Fetzer 1990). According to the pragmatic approach of Peirce, meaning is not an infused concept, but a power to engender interpretants (effects on interpreters). According to Peirce's pragmatic model of sign, meaning is a, context-sensitive (situated), interpreter-dependent, materially extended (embodied) dynamic process. It is a social-cognitive process, not merely a static system. It emphasizes process and cannot be dissociated from the notion of a situated (and actively distributed) communicational agent. It is context-sensitive in the sense that is determined by the network of communicative events within which the interpreting agents are immersed with the signs, such that they cooperate with one another. It is both interpreter-dependent and objective because it

triadically connects sign, object, and an effect in the interpreter.

CONCLUSION

The original conception of artificial intelligent systems as symbols systems brought forth a problem known as symbol grounding problem. If symbol systems manipulate symbols, these symbols should represent something to the system itself and not only to an external observer, but the system has no way of grounding these symbols in its sensory and motor interaction history, since it does not have one. Many researchers pointed this key flaw, particularly John Searle (1980) with his Chinese Room Argument and Stevan Harnad (1990) with the definition for the problem became well know.

A direction towards modeling artificial systems as embodied and situated agents instead of symbol manipulating systems was pointed out, urging the need to implement systems that could autonomously interact with its environment and with the things it should have representations for. But the topic of symbol grounding also needs a description of how certain things come to represent other things to someone, the topic of study of semiotics. The semiotics of C.S.Peirce has been used as theoretical framework in the discussion the topic of symbol grounding problem in Artificial Intelligence. The application of his theory in dealing with the symbol grounding problem should further contribute to the development of computational models of cognitive systems and to the construction of ever more meaningful machines.

REFERENCES

Brooks, R. A. (1990) Elephants Don't Play Chess. Robotics and Autonomous Systems (6), 3-15.

Brooks, R. A. (1991a) Intelligence Without Reason. Proceedings of 12th Int. Joint Conf. on Artificial Intelligence, Sydney, Australia, August 1991, 569-595.

Brooks, R. A. (1991b) Intelligence Without Representation. Artificial Intelligence Journal (47), 139–159.

Cangelosi, A., Greco, A. & Harnad, S. (2002). Symbol grounding and the symbolic theft hypothesis. In A. Cangelosi & D. Parisi (Eds.), Simulating the Evolution of Language (chap.9). London:Sprinter.

Christiansen, M.H. & Kirby, S. (2003). Language evolution: consensus and controversies. Trends in Cognitive Sciences, 7 (7), 300-307.

Clark, A. (1997) Being There: Putting Brain, Body and World Together Again. Cambridge MA: MIT Press, 1997.

Dennett, D. (1991), Consciousness Explained. Boston : Little, Brown and Co.

Dennet, D., Haugeland, J. (1987) Intentionality. In R. L. Gregory (Ed.), The Oxford Companion to the Mind, Oxford University Press.

Harnad, S. (1990) The Symbol Grounding Problem. Physica D 42: 335-346.

Haugeland, J. (1985) Artificial Intelligence: the Very Idea. The MIT Press: Cambridge, Massachusetts.

Newell, A., & Simon, H.A. (1976). Computer science as empirical inquiry: Symbols and search. Communications of the Association for Computing Machinery, 19(3), 113-126.

Peirce, C.S. (1931-1958) The Collected Papers of Charles Sanders Peirce. Vols. 1-6, Charles Hartshorne and Paul Weiss (Eds.), Vols. 7-8, Arthur W. Burks (Ed.). Cambridge, Mass.: Harvard University Press.

Roy, D. (2005) Semiotic Schemas: A Framework for Grounding Language in Action and Perception. Artificial Intelligence, 167(1-2), 170-205.

Searle, J. (1980) Minds, brains and programs. Behavioral and Brain Sciences 3, 417-457.

Searle, J. (1983) Intentionality. Cambridge: Cambridge University Press.

Vogt, P. (2002). The physical symbol grounding problem. Cognitive Systems Research, 3(3), 429–457.

Wagner, K., Reggia, J. A., Uriagereka, J., & Wilkinson, G. S. (2003) Progress in the simulation of emergent communication and language. Adaptive Behavior, 11(1):37--69.

Ziemke, T. (1999). Rethinking Grounding. In A. Riegler, M. Peschl & A. von Stein (Eds.), Understanding Representation in the Cognitive Sciences, pages 177-190. Plenum Press: New York.

KEY TERMS

Icon: A sign that represents its object by means of similarity or resemblance.

Index: A sign spatial-temporally (physically) connected with its object.

Representation: The same as a sign.

Sign: Something that stands for something else in a certain aspect to someone.

Symbol: A sign that stands for its object by means of a law, rule or disposition.

Symbol Grounding Problem: The problem related to the requirement of symbols to be grounded in something else then other symbols, if a symbol is to represent something to an artificial system.

Symbol Systems: A system that models intelligent action as symbol manipulation alone.

ENDNOTE

[1]   See also Dennett & Haugeland 1987, Searle 1983, Jacob 2003.

Symbolic Search

S

Stefan Edelkamp
University of Dortmund, Germany

INTRODUCTION

Symbolic search solves state space problems consisting of an initial state, a set of goal states, and a set of actions using a succinct representation for state sets. The approach lessens the costs associated with the exponential memory requirements for the state sets involved as problem sizes get bigger.

Symbolic search has been associated with the term *planning via model checking* (Giunchiglia and Traverso 1999). While initially applied to *model check* hardware verification problems (McMillan 1993), symbolic search features many modern *action planning* systems (Ghallab et al. 2000).

Symbolic search algorithms explore the underlying problem graph by using functional expressions to represent sets of states and actions. Compared with the space requirements induced by standard explicit-state search algorithms, symbolic representations additionally save space by sharing parts of the state vector. Algorithm designs change, as not all search algorithms adapt to the exploration of state sets.

BACKGROUND

Binary decision diagrams or BDDs are one option for a space-efficient representation for state sets.

A BDD (Bryant 1992; see Figure 1), is a data structure to manipulate Boolean functions efficiently. BDDs are finite state machines over the alphabet {0,1} with a 1-sink that operates as an accepting state. Each internal node is labelled with the variable (index) for selecting the outgoing transition (either 1 or 0, see figure) for a given variable assignment. For evaluating a BDD, a path is traced from the root to the sinks (all paths obey the same variable ordering). What distinguishes BDDs from decision trees is the use of reduction rules, detecting unnecessary variable tests and repeating subgraphs. This leads to a unique representation, polynomial in the number of input variables for many interesting functions. The reduced and ordered BDD representation is unique; a clear benefit to the satisfiability test for Boolean formulas, which by the virtue of Cook's Theorem (1971) is an NP-hard problem
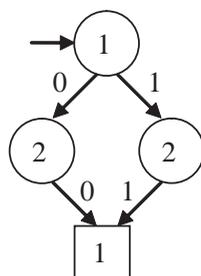
In symbolic search, BDDs accept the state vector representation. According functions are satisfied, if the state vector for the input assignment is a member of the represented set. The characteristic function can be identified with the state set it represents.

The transition relation Trans represents the actions (see Figure 2). It refers to current state variables x and next state variables x' and is satisfied, if there is an action that transforms a state vector into one of its successors. The transition relation for the entire prob-

Figure 1.

Figure 2.
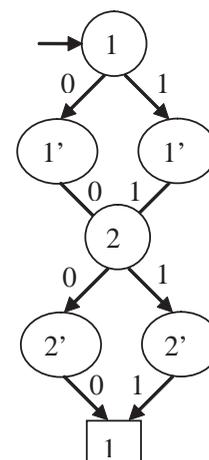
lem decomposes in the disjunction of the transition relations for singleton actions. The order of variables in the state vector is crucially influencing the size of the BDD. Unfortunately, the problem of finding the ordering that minimizes the BDD size is NP-hard (Wegener 2000). The interleaved representation for the Trans(x,x') that alternates between x and x' variables often leads to small BDDs.

The image of a state set States wrt. the transition relation Trans is computed as $\text{Image}(x') := \exists x (\text{Trans}(x,x') \land \text{States}(x))$, where x and x' are vectors of Boolean state variables. The result of this image operation is a characteristic function of all states reachable from States in one step. In order to repeat the process, x with x' have to be substituted for the next iteration by computing the *relational product* $\text{States}(x) := \exists x' ((x=x') \land \text{Image}(x'))$. In an interleaved variable ordering with alternating indices for x and x', this operation reduces to a mere textual replacement of node labels.

SYMBOLIC SEARCH ALGORITHMS

State space problems numbers of finite domain can be encoded via atomic propositions. A binary encoding is more efficient than a unary one such that most BDD libraries include finite domain variable support. For basic calculus, relations are pre-computed. For example, the binary relation Inc(a,b) for a+1=b is the disjunction of all possible value assignments of a to j and all possible value assignments of b to j+1 for all j counted from 1 to the domain size minus 1. For constructing the ternary relation Add(a,b,c), denoting a+b=c, the enumeration of all possible assignments for a, b and c is less efficient than computing the term $\text{Add}(a,b,c) := (b=0 \land a=c) \lor \exists b',c' (\text{Inc}(b',b) \land \text{Inc}(c',c) \land \text{Add}(a,b',c'))$ recursively. Starting with the first clause the second clause is applied until convergence.

Symbolic Breadth-First Search

In iteration i of the symbolic variant of breadth-first-search the set of states States[i] reachable from the initial state s in i steps is computed. The search is initialized with States[0] set to the initial state set. In order to terminate the search the algorithm checks, whether or not a state is represented in the intersection of the set States[i] with the set of goal states. Since States[0],…,States[i-1] have been computed without

success, given a non-empty intersection, i is the optimal solution length. To avoid an infinite search behaviour in case of the absence of a solution, Reach = States[0] $\lor$…$\lor$ States[i-1] is omitted from States[i] by setting States[i] := States[i] $\land \neg$Reach before updating Reach := Reach $\lor$ S[i]. For some problem classes (like undirected or acyclic graphs) the *duplicate elimination scope* {0,…,i-1} can be reduced to a limited number of breadth-first search levels.

By keeping the intermediate BDDs contained in the memory, a legal sequence of states linking the initial state to any goal state g in States[i] $\cap$ G is a successful solution. The state on an optimal path to a goal g in layer i must be located in the second last breadth-first search layer i-1. All states that are contained in the intersection of the predecessors of the goal g are and States[i-1] are reachable in an optimal number of steps and reach the goal in one step. Any of these states can be chosen to continue *solution reconstruction*. Eventually the initial state is found. If layers have been eliminated to recover main memory, divide-and-conquer solution reconstruction methods are required (Jensen et al. 2006). Variants of symbolic breadth-first search compute cost-optimal solutions subject to general cost functions (Edelkamp 2006).

Backward breadth-first search exploits the relational representation for the actions to compute the *preimage* according to the formula $\text{Preimage}(x) := \exists x' (\text{States}(x') \land \text{Trans}(x,x'))$. Consequently, the search starts with the goal state set and iterates until it hits the start state. Bidirectional symbolic breadth-first search executes concurrent iterations of forward and backward breadth-first search until the two search frontiers meet.

Symbolic Dijkstra's Single Source Shortest Paths Algorithm

Action costs are a natural search concept. In many applications, costs can only be bounded integers. Examples for such discrete cost actions are *macros* as exploited in the macro-problem solver by Korf (1985).

Let the *weighted transition relation* Trans(c,x,x') evaluate to 1, if the step from x to x' has cost $c \in \{1,…,C\}$, encoded in binary. The symbolic version of Dijkstra's single-source shortest paths algorithm (1959) then works as follows. The priority relation Queue(f,x) is initialized with the representation of the start state and f-value 0. Until a goal state is reached, in each iteration, the algorithms determines the minimum f-value

min, the relation Min(x) of all states in the priority queue with value min, and the relation Rest(f,x) of the remaining set of states Queue(f,x) \ Min(x). The transition relation Trans(c,x,x') is then applied to Min to determine the relation for the successor state set. To attach the new values f=min+c to this set, relation Add mentioned above applies. Finally, the priority relation for the next iteration is obtained by intersecting the evaluated successor set with the remaining queue. The algorithm mimics the execution of Dijkstra's algorithm on 1-level bucket data structure (Dial 1969). As f increases monotonically, the first goal extracted from the priority queue has optimal cost.

Symbolic Pattern Databases

State space abstraction is the key aspect for the automated design of search heuristics. Applying abstractions simplifies a problem, and exact distances in relaxed problems then serve as lower bound for the concrete base-level search. Proper abstractions preserve path existence. *Pattern databases,* introduced by Culberson and Schaeffer (1998), completely evaluate the abstract search space prior to the concrete search. The limitation for applying pattern databases in search practice is the restricted amount of (main) memory.

More than one pattern database can be combined by either taking the maximum (always applicable), or the sum of individual pattern database entries (only applicable if the pattern databases are disjoint). Disjoint pattern databases (Korf and Felner 2002) belong to the best known techniques to construct effective search heuristics. In order to select patterns automatically, Edelkamp (2000) as well as Haslum et al. (2005) use greedy *pattern packing* to divide the state vector into disjoint parts for constructing pattern databases that respect a pre-specified memory limit.

Symbolic pattern databases (Edelkamp 2002) are functional pattern databases for later use either in symbolic or explicit-state heuristic search. Different to a posterior compression of the state set (Felner et al. 2007), the construction itself works on a compressed data structure. Symbolic pattern databases are relations of pairs (f,x), which are satisfied if the heuristic estimate of a states encoded in x matches the heuristic value encoded in f. Such relations can be represented as a BDD for the entire problem space or kept partitioned in form of breadth-first search layers Heur[0], … , Heur[k]. This list is initialized with the abstracted goal and, as long as there are newly encountered states, the set of predecessors with respect to the abstract transition relation is generated. For constructing symbolic

Figure 3.
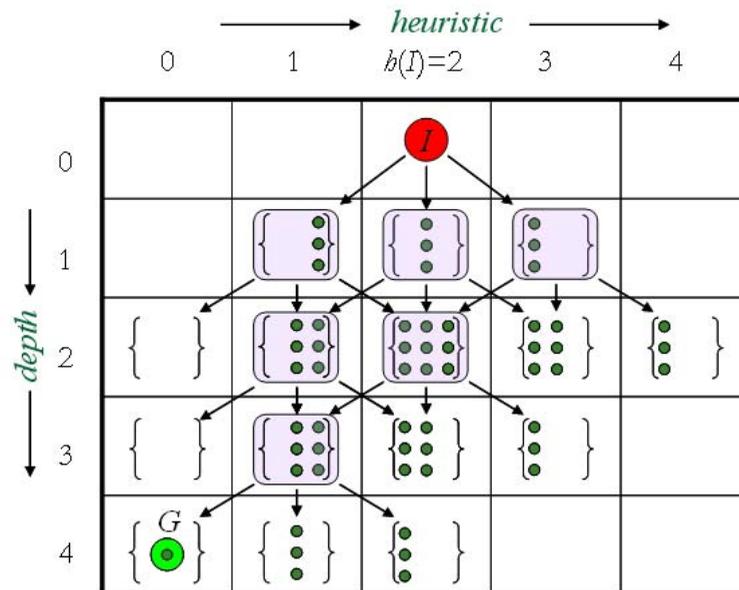
shortest-path pattern databases, Dijkstra's algorithm can be adapted.

Symbolic Version of A*

Given a consistent heuristic h with c(n,n')-h(n)+h(n') ≥ 0 for all states n and n', A* (Hart et al. 1968) is in fact a variant of Dijkstra's algorithm using an initial offset f(s):=h(s) and a refined update f(n') := min{f(n'),f(n)+c(n,n')-h(n)+h(n')}.

BDDA* integrates symbolic and A* search (Edelkamp and Reffel 1998). It determines the successors of the set of states with minimum f-value in one evaluation step. Optimality and completeness of BDDA* are inherited from explicit-state A*. Starting from solving single-agent challenges BDDA* has been applied to hardware verification (Reffel and Edelkamp 1999) and planning problems (Edelkamp 2002). The efficiency has been reproduced by Hansen et al. (2002) and Qian and Nymeyer (2003).

To avoid finite domain arithmetics with BDDs, Jensen et al. (2002) has suggested a two-dimensional layout of state sets, one for each possible g- and h-value pair (see Figure 3). The advantage is that each state set already has the g- and the h-value attached to it, and the computation of the f-values for the set of successors are no longer needed. In the extension of BDDA* to weighted actions all successors of the set of states with minimum f-value, current cost g and action cost c can be determined individually.

For computing the image, constructing a monolithic relation Trans is not mandatory. Given that sub-relations Trans[a] are linked to every action a ∈ {1,…,k} the image of state set partitions into {∃x (Trans[1](x,x') ∧ States(x))} ∨ … ∨ {∃x (Trans[k](x,x') ∧ States(x))}.

FUTURE TRENDS

Symbolic search algorithms with BDD are also effective in solving non-determinstic search, and probabilistic search problems, see initial work by Cimatti et al. (1998) or Hoey et al. (1999). Recent trends include temporally extended goals as analyzed by Lago et al. (2002).

In order save main memory, all but the currently expanded BDDs can be flushed to disk. As both explorations work on sets of states, a combination of disk-based search (to save RAM for the exploration) and symbolic pattern databases (to save RAM for the

estimate) turns out to be very effective. Distributed search may additionally save memory on the individual computing node. Successful explorations with up to 16 GB main memory and 3 TB disk space have been reported by Edelkamp and Jabbar (2007).

More advanced symbolic data structures can also cover infinite state sets. The exploration algorithms share similar algorithmic principles but have to be adapted. A recent proposal by Borowski and Edelkamp (2006) considers symbolic search in infinite-state systems with automata theory, where state sets and actions are represented as minimized finite state automata. The search repeatedly applies specialized image operators to compute the automata for the successor sets.

Another important future application area for symbolic search is the classification in general game play (Love et al. 2006), e.g. in the area of two-player games. First algorithms have been provided by Edelkamp and Kissmann (2007), which compute strategies in form of BDDs, assuming optimal play. Once computed, BDDs serve as finite-state controllers.

CONCLUSION

Symbolic search is an apparent option for the space-efficient traversal of state spaces bypassing the explicit memory-consuming representation of the state sets. The essentials of symbolic exploration in finite state systems with state sets that are represented as Boolean functions in form of BDDs have been presented, and various algorithms as well as their refined implementations including symbolic uni- and bidirectional breadth-first, single-source shortest paths, as well as heuristic search with pattern databases have been discussed.

REFERENCES

B. Borowski and S. Edelkamp (2007). *Optimal Infinite-State Planning with Presburger Automata.* Technical Report, 815, University of Dortmund.

R. E. Bryant (1992). *Symbolic boolean manipulation with ordered binary-decision diagrams.* ACM Computing Surveys, 24(3):142–170.

S. A. Cook (1971). *The complexity of theorem-proving procedures.* ACM Symposium on Theory of Computing (STOC), 151-158

S

A. Cimatti, M. Roveri, and P. Traverso (1998). Automatic OBDD-based generation of universal plans in non-deterministic domains. In National Conference on Artificial Intelligence (AAAI), 875–881.

R. B. Dial (1969). Shortest-path forest with topological ordering. Communications of the ACM, 12(11): 632–633.

E. W. Dijkstra (1959). A note on two problems in connexion with graphs. Numerische Mathe-matik 1:269–271.

S. Edelkamp and F. Reffel (1998). *OBDDs in heuristic search*. In German Conference on Artificial Intelligence (KI), 81–92.

S. Edelkamp (2001). Planning *with pattern databases.* In European Conference on Planning (ECP), 13–24.

S. Edelkamp (2002). *Symbolic pattern databases in heuristic search planning.* In Conference on Artificial Intelligence Planning and Scheduling (AIPS), 274–293.

S. Edelkamp and S. Jabbar (2007). *Pushing the Limits for Planning Pattern Databases.* Technical Report, 816, University of Dortmund.

S. Edelkamp, and P. Kissmann (2007). *Symbolic Exploration for General Game Playing in PDDL*, ICAPS-Workshop on Planning in Games.

A. Felner, R. E. Korf, R. Meshulam and R. Holte (2007). Compressed Pattern Databases. Journal of Artificial Intelligence.

M. Ghallab, D. Nau, and P. Traverso (2004). *Automated Planning: Theory & Practice.* Morgan Kaufmann.

F. Giunchiglia and P. Traverso (1999). *Planning as Model Checking*. European Conference on Planning (ECP), 1-19.

E. A. Hansen, R. Zhou, and Z. Feng (2002). *Symbolic heuristic search using decision diagrams.* In Symposium on Abstraction, Reformulation and Approximation (SARA), 83–98.

P. Haslum, B. Bonet, and H. Geffner (2005). *New admissible heuristics for domain-independent planning.* In National Conference on Artificial Intelligence (AAAI), 1163–1168.

P. E. Hart, N. J. Nilsson, and B. Raphael (1968). *A formal basis for heuristic determination of minimum path cost*. IEEE Transactions on on Systems Science and Cybernetics, 4:100–107.

J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier (1999). SPUDD: Stochastic planning using decision diagrams. In Conference on Uncertainty in Articial Intelligence (UAI), 279–288.

R. M. Jensen, R. E. Bryant, and M. M. Veloso (2002). SetA*: *An efficient BDD-based heuri-stic search algorithm.* In National Conference on Artificial Intelligence (AAAI), 668–673.

R. E. Korf (1985). Macro-operators: *A weak method for learning.* Artificial Intelligence 26: 35–77.

R. E. Korf and A. Felner (2002). Chips Challenging Champions: Games, Computers and Artificial Intelligence. Elsevier. *Chapter: Disjoint Pattern Database Heuristics*, 13–26.

D. Lago and M. Pistore and P. Traverso (2002). *Planning with a Language for Extended Goals.* National Conference on Artificial Intelligence (AAAI), 447-454.

N. C. Love, T. L. Hinrichs, and M.G. Genesereth (2006). *General game playing: Game Description language specification*. Technical Report LG-2006-01, Stanford Logic Group.

K. McMillian (1993). *Symbolic Model Checking.* Kluwer Academic Press.

K. Qian and A. Nymeyer (2003). *Heuristic search algorithms based on symbolic data structures*. In Australian Conference on Artificial Intelligence (ACAI), 966–979.

K. Qian and A. Nymeyer (2004). *Guided invariant model checking based on abstraction and symbolic pattern databases.* In Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 497–511.

F. Reffel and S. Edelkamp (1999). Error detection with directed symbolic model checking. In World Congress on Formal Methods (FM), 195–211.

I. Wegener (2000). *Branching programs and binary decision diagrams - theory and applications.* SIAM Monographs on Discrete Mathematics and Applications.

KEY TERMS

Action Planning: Refers to a world description in logic, where a number of atomic propositions describe what can be true or false in each state of the world. By applying operators to a world, one arrives at another world, where different atoms might be true or false. Usually, only few atoms are affected by an action, and most of them remain the same.

Duplicate Elimination Scope: The number of layers that a back edge in a breath-first (or best-first) search graph can cross. It is an important parameter for the design of memory-limited frontier search algorithms and has application to improve the efficiency of both symbolic and disk-based search.

Model Checking: For a system model together with a formal description of a property, model checking is a push-button decision procedure. In case the desired property is not satisfied by the model, it returns a counter-example in form of a trace. Among the options for the specification of the model, there are Kripke structures and labelled transition systems. Valid choices for property specifications are linear and branching time logics, or the propositional μ-calculus.

Pattern Database: Given that state in a search problem is described as a vector of state variables, pattern variables denote a subset of them. They define an abstraction such that any path in the concrete state space induces a path in the abstract one. A pattern is a specific assignment of values to the pattern variables. A pattern database completely evaluates the abstract search space prior to the base level search in form of a lookup table indexed by the abstract containing the shortest goal distance.

Pattern Packing: Solves the pattern selection problem for constructing pattern database search heuristics. One bin represents a container for the abstract state space and approximates the memory usage for pattern database construction. Multiple bins apply for disjoint pattern database construction. In difference to standard bin packing, the effect of the selection of patterns is multiplicative.

Relational Product: Specialized procedure that combines conjunction and variable quantification in one specialized BDD operation.

Synthetic Neuron Implementations

Snorre Aunet
University of Oslo, Norway & Centers for Neural Inspired Nano Architectures, Norway

INTRODUCTION

Many different synthetic neuron implementations exist, that include a variety of traits associated with biological neurons and our understanding of them. An important motivation behind the studies, modelling and implementations of different synthetic neurons, is that nature has provided the most efficient ways of doing important types of computations, that we are trying to mimick.

Whether it is Artificial Neural Networks (ANNs) or other mixed signal systems, technology has always evolved in the direction of lower energy per unit computation (Mead, 1990). Simple Neuron models as threshold elements, or perceptrons, are promising candidates for implementing future signal processing systems, including CMOS and SET (Schmid & Leblebici, 2003), (Beiu & Ibrahim, 2007).

In this article a small number of published subthreshold, ultra low power, perceptrons / threshold elements are compared regarding power consumption, operational speed and defect tolerance. The "mirrored" gate operating in subthreshold and combined with redundancy, might be an interesting candidate for implementing artificial neural networks as well as other mixed-signal processing circuitry.

Previously unpublished results demonstrate the mirrored gate producing appropriate binary outputs at 180 mV supply voltage, even when a transistor was cut off the supply voltage, for a redundancy factor of 2, using shorted outputs, as in (Aunet & Hartmann, 2003).

BACKGROUND

CMOS has been the dominant technology for implementing signal processing systems for decades, and will probably live alongside other nanotechnologies for a long time (ITRS, 2005). Due to needs for low power operation for about any future signal processing technology and that CMOS and similar technologies probably will be mainstream for the foreseeable future, the scope of this paper is limited to simple CMOS, ultra low power circuit topologies. Subthreshold circuits (Swansson & Meindl, 1972), using a supply voltage below the inherent threshold voltage of the transistors, consume less power than other low power circuits (Soeleman, Roy & Paul, 2001). Therefore we look at subthreshold neuron ("perceptron") implementations in this paper, and concentrate on different metrics including circuit complexity, operational speed, power consumption and defect tolerance.

Reducing the power supply voltage through using ever more modern CMOS technologies and subthreshold operation reduces the number of inputs one could use for the threshold elements, depicted in Figure 1 (Aunet, 2002). Also, since only 2 inputs is optimal to implement any arbitrary neural network (Beiu & Makaruk, 1998) we have restricted the treatment to basic building blocks having a maximum fan-in of 3.

The first simple mathematical model of the biological neurons, published by McCulloch and Pitts in 1943, calculates the sign of the weigthed sum of inputs. Sometimes such circuits are called threshold logic gates or threshold elements, illustrated in Figure 1. Such perceptrons may be used to implement Neural Networks as well as digital signal processing. For a review on a wide range of VLSI implementations the reader might confer (Beiu, Avedillo & Quintana, 2003).

Figure 1. The binary output, Y, depends on if the weighted sum of inputs X1, X2,…,Xn exceeds a certain Threshold, T.
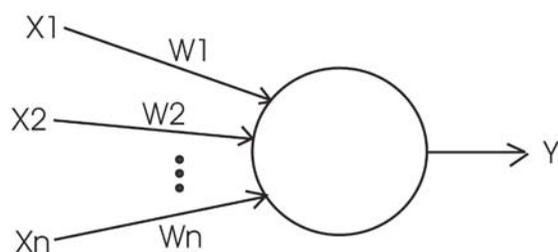
ULTRA LOW POWER NEURONS, SPEED AND RELIABILITY

The main focus is on different subthreshold ultra low power perceptrons and how they compare regarding power consumption, operational speed and reliability.

MOS Transistors in Subthreshold

For an NMOS transistor in subthreshold we have (Andreou, Boahen, Pouliquen, Pavasovic, Jenkins & Strohbehn, 1991):

$$I_{ds,n} = I_0\, e^{(\kappa Vgs/Vt)}\, e^{((1-\kappa)Vbs/Vt)}(1-e^{(-Vds/Vt)}+V_{ds}/V_0)$$

$I_{ds,n}$ expresses the current from drain to source. I_0 is the zero-bias current where the pre-exponential constants have been absorbed. This includes the channel width ("W") and the length ("L") of the MOSFET structure. V_{gs} is the gate-to-source potential, V_{ds} the drain-to-source potential and Vbs the substrate-to-source potential.

V_0 is the Early voltage, which is proportional to the channel length. κ gives the effectiveness for which the gate potential is controlling the channel current. It is often approximately 0.7-0.75 (Andreou, Boahen, Pouliquen, Pavasovic, Jenkins & Strohbehn, 1991). The thermal voltage is expressed as $V_t=kT/q$. $V_t = 25.8$ mV at room temperature.

A similar equation apply to PMOS transistors, but with opposite polarities. Exponential relationships between voltages between several nodes and the current level mean that subthreshold circuits also have operational speed and power consumption that are extremely dependent on the supply voltage, V_{dd}. For example when operated at 10 kHz a subthreshold circuit used four orders of magnitude less than a regular strong inversion circuit implementing the same function (Soeleman, Roy & Paul, 2001).

Low Fan-In Subthreshold Threshold Element ("Neuron") Circuit Implementations

Recently published circuits are shown in Figure 2. The "mirrored gate" is a static CMOS solution (Beiu, Aunet, Nyathi, Rydberg III & Djupdal, 2005), based on (Hampel D., Prost K. J. & Scheinberg N. R., 1974). The floating-gate solution P3N3 (Aunet, 2002) might not go well along with future standard CMOS due to gate leakage, while the "ijcnn" (Aunet, Oelmann, Abdalla & Berg, 2004) and "stacked" (Aunet, Berg & Beiu, 2005) gates are CMOS.

Metrics Regarding Power Consumption and Maximum Operational Speed

Recently published results are shown in Figure 3 (Granhaug & Aunet, 2006). The "mirrored", "ijcnn" and "stacked" gates were used for implementing 1-bit addition, Full Adders, in a 90 nm CMOS technology, and compared to a standard CMOS implementation (upper right corner in Figure 4).

Figure 2. Experimental setup for statistical simulation of 1-bit adder. From left to right they are called "mirrored", "P3N3", "IJCNN" and "Stacked" threshold elements.
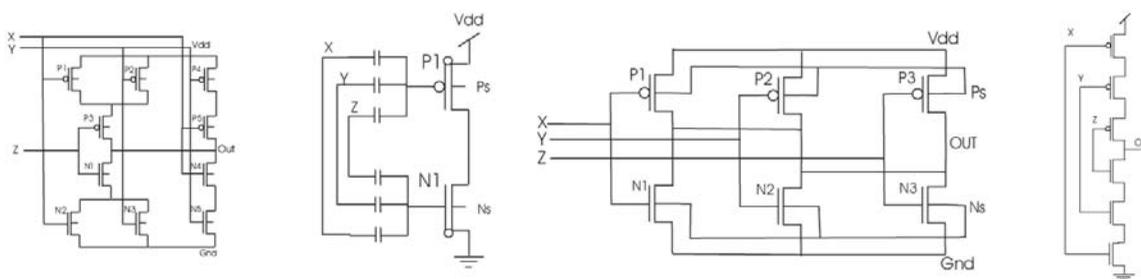
In the upper left corner one can see that exploiting the "ijcnn" gate lead to the highest power consumption, while the "stacked" gate gave the lowest power consumption among the four implementations. Spice simulations were performed using a Cadence SW environment. The "mirrored" gate implementation resulted in a power consumption of 1160 pW, while the standard CMOS implementation had a slightly lower power consumption of 932 pW according to the simulations.

Regarding the maximum operational speed, shown upper right in Figure 3, the implementation based on the "stacked" gate (Figure 2.) could not compete, while the standard, "mirrored" and "ijcnn" gate implementations had delays of 162 ns, 159 ns and 174 ns, respectively. That the "mirrored" gate was slighly faster than the "ijcnn" gate is different to the findings in another publication (Aunet & Berg, 2005), where the "ijcnn" gate lead to a delay of 5.2 us, nearly twice as fast as the "mirrored" gate with it's 9.15 us, implemented in a 120 nm CMOS technology and simulated operating at a supply voltage of 100 mV.

Manufacturability including Defect Tolerance

According to the ITRS Roadmap (ITRS, 2005), *reducing the overall power consumption and increas-*

ing the manufacturability are the two most important among five grand challenges for future nanoelectronics. Included in manufacturability are the possibilities to cope with a drastically increasing number of on-chip defects, as well as parameter variations. We have therefore included some data on the yield, meaning the expected percentage of circuits working under statistical variations in process parameters (Aunet & Otnes Berge, 2007).

A redundancy scheme for defect tolerance (Aunet & Hartmann, 2003) was used, exploiting shorted driven nodes and a redundancy factor of only 2 (R = 2), instead of redundancy factor 3 and a majority voter. Full Adders based on the three threshold elements depicted in Figure 2 were used, as well as the standard Full Adder ("FA") in the upper right corner of Figure 4. A typical digital chip will fail if even a single transistor is defective (Mead, 1990). The results shown in the lower left corner of Figure 3 (Aunet & Otnes Berge, 2007) indicate that the solution based on R = 2 and the standard CMOS FA, as well as the solution based on the "mirrored" gate, should have a supply voltage above 150 mV if the implemented circuitry should expect a 100 % yield under the 90 nm CMOS production process variations.

When small size transistors were used here, for all four solutions, the "ijcnn" and "stacked" gates could not be expected to give a satisfying yield at supply voltages

Figure 3. Comparisons regarding power consumption, delay, tolerance to certain defects and complexity of circuitry when different threshold elements are used for 1-bit addition. A standard CMOS implementation is included for comparisons.
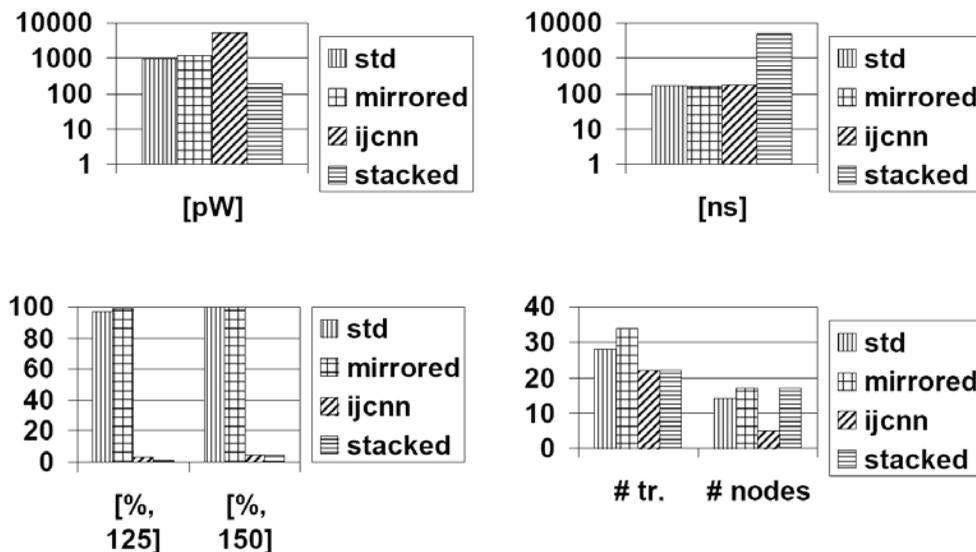
of 125 mV and 150 mV, since the gross majority of the circuits would not be expected to work.

Interconnect Challenges, and Neurons Leading to Simpler Circuitry

The number of defects in future nano technologies, including CMOS, will increase drastically. (Fortes, 2003), (ITRS, 2005). Defect tolerance must be part of about any system design (Lehtonen, Plosila & Isoaho, 2005). This include defects in interconnect and contacts. Few internal nodes generally reduce the amount of wiring and interconnect. From this viewpoint it might be preferable to have relatively few (driven) nodes as well as transistors. In this respect the "ijcnn" circuit is favorable among the four, as may be seen from figure 3 (Granhaug & Aunet, 2006).

Boolean functions that can be realized by neurons are called linear threshold functions, and they can implement any Boolean function (Siu & Bruck, 1990). Using the linear threshold elements (perceptrons) for implementing such functions may save wires, contacts and transistors, as the number of gates necessary to implement these important functions are growing linearly when using threshold elements, but exponentially if Boolean logic is used (Siu & Bruck, 1990). And: Regardless of the number of bits, Boolean gates will never lead to a lower number of gates than threshold elements.

This is also illustrated in figure 4, showing that the implementations (lower) depending on threshold elements / perceptrons are simpler and more regular than the traditional implementations, especially the floating-gate implementation based on pure Boolean logic, in the upper left corner.

As device and interconnect parameter variations imposes more problems in future nanoscale technolgies it can also be of importance to have increasingly regular stuctures on chip to reduce for example dopant fluctuations that degrade performance due to transistor threshold voltage variations. In this respect perceptron implementations sometimes can be favorable, as illustrated in Figure 4, where the two lowermost schematics have a considerably higher regularity than

Figure 4. Four schematics showing circuits implementing SUM' and CARRY' for binary addition
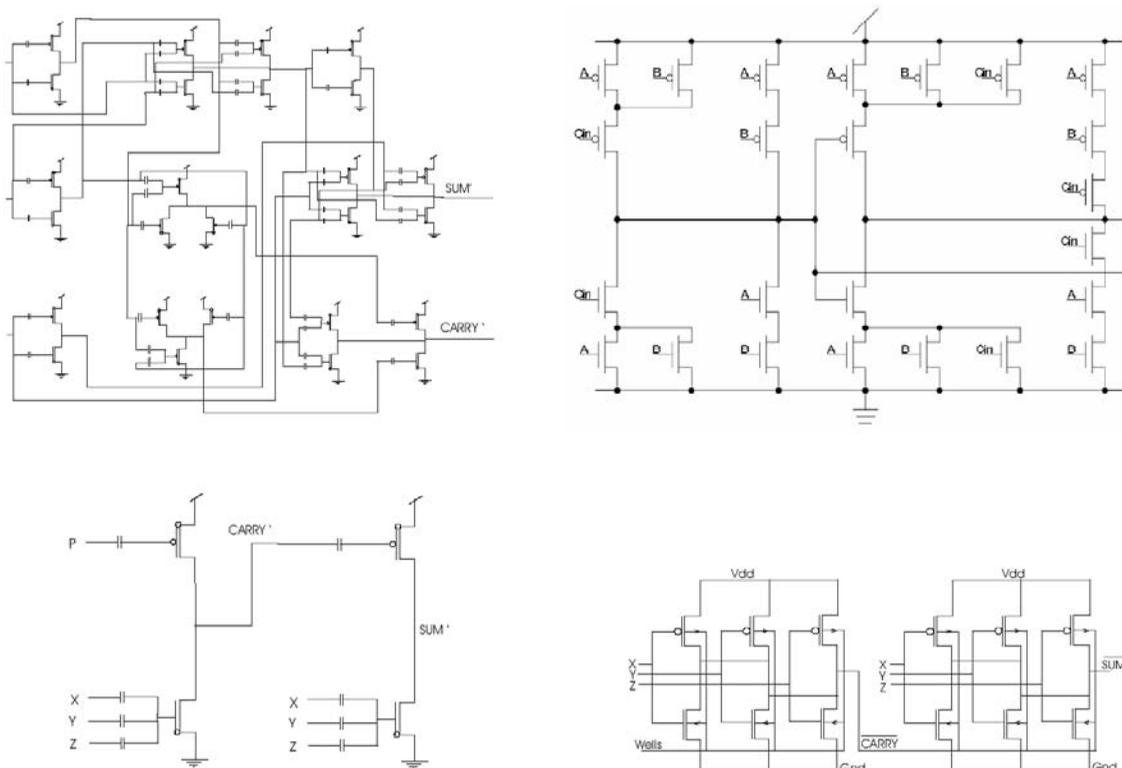
the two others. This will reflect on the layout and the physical parameters.

"Mirrored" Gate in 90 nm CMOS Computing CARRY ' Function at V_{dd} = 180 mV

Figure 5 is showing measured results from an implementation of the "mirrored" gate computing the minority 3 function at 180 mV, both with and without a stuck-open fault, for a redundancy factor of R = 2. 16 binary input vectors $[X, Y, Z] = 000, 000, 001, 001, 010, 010, \ldots, 111, 111$ were applied. The circuits computed the correct logic levels in all cases, producing a low output if, and only if, 2 or 3 out of three inputs were high. A low signal is less than 0.25 times the supply voltage of 180 mV, while a high signal should be at least 0.75 times the supply voltage.

This supply voltage is comparable to the low voltage of 175 mV published in (Miyazaki, Kao & Chandrakasan, 2002). Reducing the power supply voltage is the most direct and dramatic means of reducing the power consumption.

Discussion

Though the "stacked" gate had the lowest power consumption, it is the slowest. It does not have a low number of internal nodes and wires, which in addition to a relatively low yield according to the statistical Monte Carlo simulations might not make it the best candidate among the 4 implementations.

The "ijcnn" gate competes well regarding operational speed, but is the most power-hungry. On one hand it has the simplest topology, but the standard CMOS implementation as well as the "mirrored" gate show far better yields.

The "mirrored" gate implementation of the Full Adder competes well with the standard CMOS implementation overall (Granhaug & Aunet, 2006). These solutions might be interesting for further comparisons, as recent results have pointed out the "mirrored gate" as an interesting alternative, examplified by the following title; *Why Inverters and Small Fan-In Voters are The Most Promising Gates for Future Nanoelectronics* (Beiu & Ibrahim, 2007).

Figure 5 . Measured results
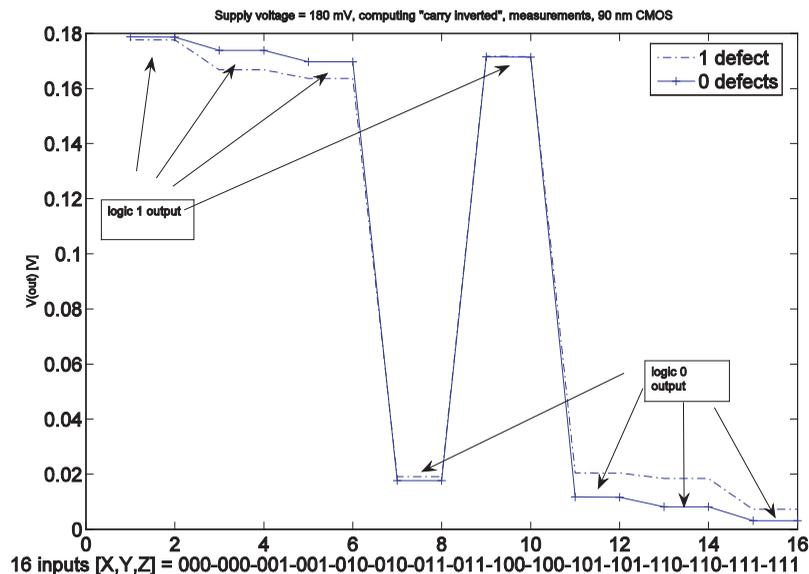
FUTURE TRENDS

The assumption that a system is composed largely of correctly functioning units is no longer true in emerging nanoelectronics, and reducing the overall power consumption is also among the grand challenges for the future. The low fan-in perceptrons, also called *voters*, or *minority gates*, might be very useful candidates for future nanoelectronics, which has been recently stated (Beiu & Ibrahim, 2007), and is not in disagreement with results presented here. Defect tolerant subthreshold perceptron circuits exploiting majority gates, as presented here, may thus become very useful.

Perceptrons may also prove to be useful building blocks for other future nanotechnologies, including SET (Beiu, Avedillo & Quintana, 2003).

CONCLUSION

We have argued that future needs for low power consumption and defect tolerance make subthreshold neuron implementations useful for artificial neural network. From the results and discussion presented here we conclude that the "mirrored" gate may be particularly useful in the mentioned respects. It may be combined with redundancy for defect tolerance and increased manufacturability, which could be useful for future nanotechnologies.

The ability of the "mirrored" gate to function under the presence of defects, when exploiting redundancy, was demonstrated by chip measurements for a 90 nm CMOS technology.

REFERENCES

Andreou A. G., Boahen K. A., Pouliquen P. O., Pavasovic A., Jenkins R. E., Strohbehn K. (1991), Current-Mode Subthreshold MOS Circuits for Analog VLSI Neural Systems, *IEEE Transactions on Neural Networks.* 205-213

Aunet S. (2002), *Real-time reconfigurable devices implemented in UV-light programmable floating-gate CMOS*, Ph.D. thesis, Norwegian University of Science and Technology, 2002, ISBN 82-471-5447-1, ISSN 0809-103X

Aunet S. & Berg Y. (2005), Three Sub-fJ Power-Delay-Product Subthreshold CMOS Gates, *Proc. IFIP VLSI-SOC,* 465-470

Aunet S., Berg Y. & Beiu V. (2005), Ultra Low Power Redundant Logic Based on Majority-3 Gates *Proc. IFIP VLSI-SOC,* 553-558

Aunet S. & Hartmann M. (2003), Real-time Reconfigurable Threshold Elements and Some Applications to Neural Hardware. *Proc. 5th International Conference on Evolvable Systems, LNCS.* 365-376

Aunet S., Oelmann B., Abdalla S. & Berg Y. (2004), Reconfigurable subthreshold CMOS perceptron. *Proc. IEEE Int.'l Conf. on Neural Networks,* 1983-1988

Aunet S. & Otnes Berge H. K. (2007), Statistical Simulations for Exploring Defect Tolerance and Power Consumption for 4 1-bit Addition Circuits. *Proc. 9th International Work-Conference on Artificial Neural Networks, LNCS.* 455-462

Beiu V., Aunet S., Nyathi J., Rydberg R. R. III & Djupdal A. (2005), On the advantages of serial architectures for low-power reliable computations. *Proc. IEEE Int.'l Conference on Application Specific Systems, Architectures and Processors*, 276-281.

Beiu V., Quintana J. M., Avedillo M. J. (2003), VLSI Implementations of Threshold Logic – a Comprehensive Survey, *IEEE Transactions on Neural Networks.* 1217 – 1243

Beiu V., Ibrahim W. (2007), Why Inverters and Small Fan-In Voters are The Most Promising Gates for Future Nanoelectronics. *Proc. 16th Int.'l Workshop on Post-Binary ULSI Systems, Oslo*.

Beiu V. & Makaruk H. E. (1998), Deeper Sparser Nets Can Be Optimal, Neural Proc. Letters, December 1998, 201 - 210

Fortes J. A. B., Future Challenges in VLSI System Design, *Proc. IEEE Computer Society Annual Symposium on VLSI*, 5-7.

Granhaug K. & Aunet S. (2006), Six Subthreshold Full Adder Cells Characterized in 90 nm CMOS Technology. *Proc. Design and Diagnostics of Electronic Circuits and Systems,* 25-30

Hampel D., Prost K. J. & Scheinberg N. R. (1974), *"Threshold Logic using Complementary MOS Device"* U.S. Patent 3900 742, June 24.

(ITRS, 2005) *International Roadmap for Semiconductors,* 2005 Edition - Executive Summary – available: http://www.itrs.net

Lehtonen T., Plosila J. & Isoaho J. (2005), *On Fault Tolerance Techniques towards Nanoscale Circuits and Systems* Turku Center for Comp. Sci., Tech. Rep.

Mead C. A. (1990), Neuromorphic Electronic Systems, *Proceedings of the IEEE*, 1629-1636

Miyazaki M., Kao J. & Chandrakasan A. P. (2002), A 175 mV Multiply-accumulate unit using an adaptive supply voltage and body bias (asb) architecture. *Proc. IEEE International Solid-State Circuits Conference*, 58-444

Schmid A. & Leblebici Y. (2003), Robust Circuit and System Design Methodologies for Nanometer-Scale Devices and Single-Electron Devices. *Proc. Third IEEE Conference on Nanotechnology*. 516-519

Siu K. Y. & Bruck J. (1990), Neural Computation of Arithmetic Functions, *Proceedings of the IEEE*, 1669-1675

Soeleman H., Roy K. & Paul B. C. (2001), Robust Subthreshold Logic for Ultra Low Power Operation, *IEEE Transactions on Very Large Scale Integration Systems*, 90-99

Swanson R. & Meindl J. D. (1972), Ion-Implanted Complementary MOS Transistors in Low-Voltage Circuits. *Proc. IEEE Int.'l Solid-State Circuits Conf.* 192-193

Wong, B., Mittal, A., Cao Y. & Starr G. W. (2004), *Nano-CMOS Circuit and Physical Design.* ISBN: 978-0-471-46610-9

KEY TERMS

Full Adder: Circuit that produces the binary sum and carry when adding two binary numbers.

Mismatch: Ideally identically constructed elements on an integrated circuits have a mismatch when they differ in their physical properties after production of the chip.

Minority-3 Gate: A minority 3 gate outputs a logic "0" signal if, and only if, 2 or 3 out of it's three binary inputs are "1".

Monte Carlo Simulations: Computer simulations basing the results on statistical distribution of parameters.

Nanoscale CMOS: CMOS technologies where dimensions smaller than 100 nm is critical to the functioning of the produced chip.

Neuron: Electrically excitable cells in the nervous system that process and transmit information.

Parameter Variations: Parameters describing physical traits of integrated circuits may have variations due to mismatch, for example the threshold voltages of transistors.

Perceptron: Type of artificial (feedforward) Neural Network.

SET: Single Electron Transistor.

Yield: In this paper the term yield refers to the ratio of functional circuits to the total number of simulated circuits. Often yield refers to the ratio of functional chips to the total number of manufactured chips.

Teaching Machines to Find Names

Raymond Chiong
Swinburne University of Technology, Sarawak Campus, Malaysia

INTRODUCTION

In the field of Natural Language Processing, one of the very important research areas of Information Extraction (IE) comes in Named Entity Recognition (NER). NER is a subtask of IE that seeks to identify and classify the predefined categories of named entities in text documents. Considerable amount of work has been done on NER in recent years due to the increasing demand of automated texts and the wide availability of electronic corpora. While it is relatively easy and natural for a human reader to read and understand the context of a given article, getting a machine to understand and differentiate between words is a big challenge. For instance, the word 'brown' may refer to a person called Mr. Brown, or the colour of an item which is brown. Human readers can easily discern the meaning of the word by looking at the context of that particular sentence, but it would be almost impossible for a computer to interpret it without any additional information.

To deal with the issue, researchers in NER field have proposed various rule-based systems (Wakao, Gaizauskas & Wilks, 1996; Krupka & Hausman, 1998; Maynard, Tablan, Ursu, Cunningham & Wilks, 2001). These systems are able to achieve high accuracy in recognition with the help of some lists of known named entities called gazetteers. The problem with rule-based approach is that it lacks the robustness and portability. It incurs steep maintenance cost especially when new rules need to be introduced for some new information or new domains.

A better option is thus to use machine learning approach that is trainable and adaptable. Three well-known machine learning approaches that have been used extensively in NER are Hidden Markov Model (HMM), Maximum Entropy Model (MEM) and Decision Tree. Many of the existing machine learning-based NER systems (Bikel, Schwartz & Weischedel, 1999; Zhou & Su, 2002; Borthwick, Sterling, Agichten & Grisham, 1998; Bender, Och & Ney, 2003; Chieu & Ng, 2002; Sekine, Grisham & Shinnou, 1998) are able to achieve near-human performance for named entity

tagging, even though the overall performance is still about 2% short from the rule-based systems.

There have also been many attempts to improve the performance of NER using a hybrid approach with the combination of handcrafted rules and statistical models (Mikheev, Moens & Grover, 1999; Srihari & Li, 2000; Seon, Ko, Kim & Seo, 2001). These systems can achieve relatively good performance in the targeted domains owing to the comprehensive handcrafted rules. Nevertheless, the portability problem still remains unsolved when it comes to dealing with NER in various domains.

As such, this article presents a hybrid machine learning approach using MEM and HMM successively. The reason for using two statistical models in succession instead of one is due to the distinctive nature of the two models. HMM is able to achieve better performance than any other statistical models, and is generally regarded as the most successful one in machine learning approach. However, it suffers from sparseness problem, which means considerable amount of data is needed for it to achieve acceptable performance. On the other hand, MEM is able to maintain reasonable performance even when there is little data available for training purpose. The idea is therefore to walkthrough the testing corpus using MEM first in order to generate a temporary tagging result, while this procedure can be simultaneously used as a training process for HMM. During the second walkthrough, the corpus uses HMM for the final tagging. In this process, the temporary tagging result generated by MEM will be used as a reference for subsequent error checking and correction. In the case when there is little training data available, the final result can still be reliable based on the contribution of the initial MEM tagging result.

BACKGROUND

Message Understanding Conference

In 1987, the Naval Ocean Systems Center (NOSC), which is presently known as the Naval Command,

Control and Ocean Surveillance Center, initiated the first Message Understanding Conference (MUC). Subsequently, a series of MUCs had been held and designed to promote and evaluate research in IE. The evaluations achieved through these MUCs have led the research program in IE until its present state.

In 1995, goals and tasks were set up for MUC-6 to make the IE system more practical with an aim to achieve automatic performance with high accuracy. "Named Entity" was then developed to help identifying the names of persons, organizations, and geographic locations in a text. Since then, the NER tasks have become a central theme in MUC (see Chinchor, 1995 and Chinchor, 1998 for more details).

According to the specifications defined by MUC, the NER tasks generally work on seven types of named entities as listed below with their respective markup:

- PERSON (ENAMEX)
- ORGANIZATION (ENAMEX)
- LOCATION (ENAMEX)
- DATE (TIMEX)
- TIME (TIMEX)
- MONEY (NUMEX)
- PERCENT (NUMEX)

From the list above, three subtasks are derived from these seven types of named entities and assigned with three respective SGML tag elements, namely ENA-MEX, TIMEX and NUMEX. As TIMEX and NUMEX are fairly easy to predict with some effective finite state methods (Roche & Schabes, 1997), most of the current research deals only with ENAMEX which are highly variable and ambiguous.

Previous Approaches

Since MUC-6 and MUC-7, many NER systems have been proposed and proven to be successful in their targeted domains. In general, NER systems that use handcrafted rules still lead the way, with the highest F-measure score up to 96.4% achieved in MUC-6 as compared to the statistical approaches that were able to achieve 94.9% (Zhou & Su, 2002).

In rule-based approach, a set of rules or patterns is defined to identify the named entities in a text. These rules or patterns consist of distinctive word format, such as capitalization or particular preposition prior to a named entity. For instance, a capitalized string behind titles such as 'Mr', 'Dr', etc will be identified as name of a person, whereas a capitalized word after a preposition such as 'in', 'at', 'near', etc is most likely to be a location. By implementing a finite set of carefully predefined pattern matching rules, the named entities within a text could be found systematically.

There have been substantial amount of works done using the rule-based approach. One of the very well documented systems that followed the direction of this approach was the framework of the LaSIE System reported by Wakao et al. (1996). Another well-known example of rule-based system can be found in the IsoQuest's NetOwl Text Extraction System presented by Krupka and Hausman (1998). Meanwhile, Diana Maynard et al. (2001) had also built an NER system based on handcrafted rules that is able to achieve an average of 93% precision and 95% recall across diverse text types.

Statistical approach, on the other hand, works by using a probabilistic model containing features to the data which are similar to the rule-based approach. The features of the data, which could be understood as rules set for the probabilistic model, are produced by learning the resulting corpora with correctly marked named entities. The probabilistic model then uses the features to calculate and identify the most probable named entities. As such, if the annotated features of the data are truly reliable, the model would have a high probability in finding almost all the named entities within a text.

In the last decade, large amount of works in NER have been done using the statistical approach based on some very large corpora. The MEM, one of the most popular statistical models, has been applied frequently in various NER tasks. One significant account on MEM is the MENE system reported by Borthwick et al. (1998). In their system, they used four main features to identify the named entities, which they referred to as the binary features, lexical features, section features and dictionary features.

The binary features in MENE system basically deal with capitalization in the text. Meanwhile, lexical features are concerned with the lexical terms such as list of words and their types which are used with a grammar. Section features indicate a current section of the text, whereas the dictionary features make use of a broad array of dictionaries of single or multiple terms such as first names, organization names, corporate suffixes, etc. The dictionary features are similar to the gazetteers

used for rule-based systems, except that dictionaries in MENE system require no huge maintenance effort.

Nevertheless, using MENE system alone on the MUC-7 test data as reported in Borthwick et al. (1998) achieved only an F-measure of 84.22%. For MENE system to work better, Borthwick et al. combined MENE with other rule-based approaches in order to achieve superior results.

Besides Borthwick et al., Bender et al. (2003) also reported on an NER system that was able to achieve an F-measure score of 89.58% by using MEM. With an annotated corpus and a set of features, they first built a baseline named entity recognizer which was then used to extract the named entities and their contextual information from non-annotated data. The accuracy of their system was further improved with a final recognizer that made use of the trained data.

Another MEM-based system can be found in Chieu and Ng (2002). They presented a system called MEN-ERGI that made use of global information with just one classifier, and showed that their system was able to achieve performance comparable to the best machine learning-based systems in MUC-6 and MUC-7.

Apart from MEM, HMM is another well-known statistical model that has been used frequently in various NER systems. The IdentiFinder reported by Bikel et al. (1999) using a modified HMM was the best-performer on the official MUC-6 and MUC-7 test data among all the machine learning-based systems. IdentiFinder employed similar features to those of MENE system, and depended on statistics to make decision in identifying the named entities. It is different in a way that it has a complete probabilistic model that governs all decisions in classifying the named entities and models the categories of interest and the residual input that is not of interest.

The modified HMM used by IdentiFinder was subsequently adopted by Zhou and Su (2002). In their work, they were able to increase the performance of their NER system dramatically by introducing four sub-features with back-off modelling. Using the test data from MUC-6 and MUC-7, their system was able to achieve F-measure scores of 96.6% and 94.1% respectively.

Many more previous works were done using statistical models other than MEM and HMM. There are also many NER systems that use a hybrid approach by combining the statistical models with some rule-based learning techniques. One very successful example can

be found in the work of Mikheev et al. (1999), where they used substantial handcrafted rules together with MEM for partial matching. Observation on the previous approaches, however, shows that no system has ever tried to use MEM and HMM successively.

THE HYBRID APPROACH

As mentioned before, the NER system presented in this article uses two statistical models – MEM and HMM – in succession. The MEM is based on the MENE system reported by Borthwick et al. (1998) whereas the HMM is based on the IdentiFinder reported by Bikel et al. (1999). The system is built with Java using the existing implementation from the JavaNLP repository which is available at http://nlp.stanford.edu/javanlp/. For training and experimental purposes, British National Corpus (BNC) which contains texts that are diverse in terms of domain, style and genre has been chosen to be the testing corpus. This is to ensure that the proposed NER system is domain-independent and can adequately cope with a variety of text types.

Maximum Entropy

By following the guidelines from MUC-6 and MUC-7 for the definition of the NER task, every word from the corpus is tokenized and assigned to a desired category of named entity with the tag of either "person" (<PER>), "organisation" (<ORG>) or "location" (<LOC>). MEM is first used to estimate the probability of a given word being fallen into one of the three categories mentioned based on a set of features and some training data. Two special conditions are taken into consideration when a word falls at the beginning (<START>) and at the end (<END>) of a sentence. In the case when a given word does not fall into any of the desired categories, empty tag (<>) will be placed to indicate that the word belongs to none of the desired categories.

For the purpose of finding named entities, the maximum entropy estimation process uses a model that is described below to compute the conditional probability P for all tags t based on the history h, in which every feature f_i is associated with it a weighting parameter α_i:

$$\sup_{\theta \in \Omega} \sup_{x, y \in \Omega} \frac{p_\theta(y)}{q(x, y)} < \infty$$

T

It is necessary to note that the history *h* mentioned in the model refers to all the conditioning data that enable the system to make a decision on the tagging process. It comprises of all information derivable from the corpus relative to a token whose tag the system is trying to determine, may it be the word itself or the features. The product of the weightings for all features active on *h* will then be calculated, and eventually be divided by a normalization function, $Z_\alpha(h)$.

Hidden Markov Model

After the MEM walkthrough, all the tagged named entities in the testing corpus are used as training data for HMM to make the final tagging. Since there will be sufficient training after parsing through the corpus using MEM, it is not necessary for the system to use the back-off models such as those used by Bikel et al. (1999) and Zhou and Su (2002).

In this system, HMM is used mainly for global context checking, that is to check the occurrences of the same named entity in different sections of the same text document. Checking the context from the whole document is important as this will ensure the consistency of the tagged named entities and resolve some ambiguous cases. For instance, an organization's name is often abbreviated especially when it has already been mentioned somewhere in a document. By checking the global information, the abbreviation as an organization can be identified. Besides that, there are also some entities that are highly ambiguous, and their categories cannot be determined without taking the global context into consideration. The phrase 'Honda City' in sentences such as "Honda City is nice" or "Promotion for Honda City" could easily be misinterpreted as a location based on the local contextual evidence, unless there is another sentence that sounds like "I am driving Honda City".

Similar to the previously used MEM, HMM is used to compute the likelihood of words occurring within a given category of named entity. Every tokenized word is now considered to be in ordered pairs. By using a Markov chain, the likelihood of the words is calculated simply based on the previous word.

For classifying the named entities, the system finds the most likely tag *t* for a given sequence of words *w* that maximizes $P(t|w)$. The occurrences of the given events are counted throughout the whole text based on the calculation below:

$$P(t \mid t_{-1}, w_{-1}) = \frac{count(t, t_{-1}, w_{-1})}{count(t_{-1}, w_{-1})}$$

Finally, a classifier is used to correct the errors in the results derived from MEM to perform the final tagging process using HMM.

Experimental Results

The proposed system has been tested with articles from BNC based on a wide range of domains from different fields. With the successive use of MEM and HMM, it is able to maintain a desirable performance regardless of the size of training data. In overall, the system achieved F-measure scores above 95% consistently for most of the commonly used domains. A detailed description of the system and its experimental results can be found in Chiong and Wang (2006).

FUTURE TRENDS

While the preliminary results on the hybrid approach have been quite positive, the proposed system is still fairly immature. Much work needs to be done to make the performance of the system more robust. For instance, it will be interesting to see how more sophisticated features can be incorporated to improve the performance of the system. It will also be interesting to see how the system can be trained on corpora in foreign languages.

In the future, it is anticipated that the proposed approach can be valuable in various Natural Language applications. One immediate contribution can be seen in automating the arduous task of ontology building. Meanwhile, Automatic Text Summarization Systems can also be enriched by the proposed system, as named entities are able to provide clues for identifying relevant segments in text. Last but not least, the proposed approach is expected to help in building more accurate Internet search engines too.

CONCLUSION

This article presented a hybrid machine learning approach that used MEM and HMM successively. With the preliminary data training through MEM and appropriate classifier for error correction in the final recognition process through HMM, the performance of the proposed NER system can be greatly enhanced as compared to using only a single statistical model. Moreover, the system is also able to adapt to different domains without human intervention, and maintained desirable performance regardless of the size of the training corpus.

REFERENCES

Bender, O., Och, F.J., & Ney, H. (2003). Maximum Entropy Models for Named Entity Recognition. In *Proceedings of the 7th Conference on Computational Natural Language Learning* (CoNLL 2003), Edmonton, Canada, pp. 148-151.

Bikel, D.M., Schwartz, R.L., & Weischedel, R.M. (1999). An algorithm that learns what's in a name. *Machine Learning*, *34(1-3)*, 211-231.

Borthwick, A., Sterling, J., Agichten, E., & Grisham, R. (1998). Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proceedings of the 6th Workshop on Very Large Corpora* (WVLC-98), Montreal, Canada, pp. 152-160.

Chieu, H.L. & Ng, H.T. (2002). Named Entity Recognition: A Maximum Entropy Approach Using Global Information. In *Proceedings of the 19th International Conference on Computational Linguistics* (COLING 2002), Taipei, Taiwan, pp. 190-196.

Chinchor, N. (1995). MUC-6 Named Entity Task Definition (Version 2.1). *MUC-6*, Columbia, Maryland.

Chinchor, N. (1998). MUC-6 Named Entity Task Definition (Version 3.5). *MUC-7*, Fairfax, Virginia.

Chiong, R. & Wang, W. (2006). Named Entity Recognition Using Hybrid Machine Learning Approach. In *Proceedings of the 5th IEEE International Conference on Cognitive Informatics* (ICCI 2006), Beijing, China, pp. 578-583.

Krupka, G.R., & Hausman, K. (1998). IsoQuest Inc: Description of the NetOwl Text Extraction System as used for MUC-7. In *Proceedings of the 7th Message Understanding Conference* (MUC-7). Fairfax, Virgina.

Maynard, D., Tablan, V., Ursu, C., Cunningham, H., & Wilks, Y. (2001). Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing Conference* (RANLP 2001), Tzigov Chark, Bulgaria, pp. 257-274.

Mikheev, A., Moens, M., & Grover, C. (1999). Named Entity Recognition without Gazetteers. In *Proceedings of the 9th European Chapter of the Association of Computational Linguistics* (EACL 1999), Bergen, Norway, pp. 1-8.

Roche, E. & Schabes, Y. (1997). *Finite-State Language Processing*. Cambridge MA: The MIT Press.

Sekine, S., Grishman, R., & Shinnou, H. (1998). A Decision Tree Method for Finding and Classifying Names in Japanese Texts. In *Proceedings of the 6th Workshop on Very Large Corpora* (WVLC-98), Montreal, Canada, pp. 171-178.

Seon, C., Ko, Y., Kim, J. & Seo, J. (2001). Named Entity Recognition using Machine Learning Methods and Pattern-Selection Rules. In *Proceedings of the Natural Language Processing Pacific Rim Symposium* (NLPRS 2001), Tokyo, Japan, pp. 229-236.

Srihari, R.N. & Li, W. (2000). A Hybrid Approach for Named Entity and Sub-Type Tagging. In *Proceedings of the 6th Conference on Applied Natural Language Processing* (ANLP 2000), Seattle, Washington, pp. 247-254.

Wakao, T., Gaizauskas, R., & Wilks, Y. (1996). Evaluation of an Algorithm for the Recognition and Classification of Proper Names. In *Proceedings of the 16th International Conference on Computational Linguistics* (COLING 1996), Copenhagen, Denmark, pp. 418-423.

Zhou, G.D. & Su, J. (2002). Named Entity Recognition Using a HMM-based Chunk Tagger. In *Proceedings of the 40th Annual meeting of the Association for Computational Linguistics* (ACL 2002), Philadelphia, pp.473-480.

KEY TERMS

British National Corpus (BNC): A 100 million word collection of samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of current British English, both spoken and written.

Gazetteer: A list of named entities with the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

Hidden Markov Model (HMM): A statistical model for determining the hidden parameters based on the observable parameters using probability distribution in order to perform analysis and pattern recognition.

Information Extraction (IE): A process of selecting information from a dataset or text based on certain specifications using templates, and it is often delivered in the form of fragments of documents.

Machine Learning: An area of study concerning the development of techniques which allow machines or computers to improve their performance based on previous results and learning experience.

Maximum Entropy Model (MEM): A statistical model for analyzing the available information in order to determine a unique epistemic probability distribution based on partial information about the probabilities of possible outcomes of an experiment, and chooses the probabilities so as to maximize the uncertainty about the missing information.

Named Entity Recognition (NER): A subtask of IE that seeks to identify and classify named entities in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

Natural Language Processing (NLP): An area of study concerning the problems inherent in the processing and manipulation of natural language with the aim to make computers understand statements written in human languages.

T

Thermal Design of Gas-Fired Cooktop Burners Through ANN

T. T. Wong
The Hong Kong Polytechnic University, Hong Kong

C. W. Leung
The Hong Kong Polytechnic University, Hong Kong

INTRODUCTION

Recent advances in the applications of ANN have demonstrated successful cases in time series analysis, data mining, civil engineering, financial analysis, music creation, fishing prediction, production scheduling, intruder detection, etc., making them an important tool for research and development[1]. ANN and evolutionary computation(EC) techniques have been employed successfully in solving real-world problems including those with a temporal component[2]. In another work[3], a hybrid method based on a combination of evolutionary computation and neural network(NN) has been used to predict time series.

In the world of databases, various ANN-based strategies have been used for knowledge search and extraction[4]. Intelligent neural systems have been constructed with the aid of genetic algorithm-based EC techniques and these systems have been applied in breast cancer diagnosis[5]. Genetic algorithms(GA) have been applied to develop a general method of selecting the most relevant subset of variables in the field of analytical chemistry to classify apple beverages[6]. New ANN methods enable civil engineers to use computing in different ways. Besides as a tool in urban storm drainage[7], ANN and Genetic Programming(GP) have been implemented in the prediction and modelling of the flow of a typical urban basin [8]. In the latter case, it was shown that these two techniques could be combined in order to design a real-time alarm system for floods or subsidence warning in various types of urban basins. ANN models for consistency, measured by slump, in the case of conventional concrete have also been developed[9]. In a time series prediction of the quarterly values of the medical component of the Consumer Price Index(CPI), the results obtained with both neural and functional networks have been shown

to be quite similar[10]. Dimensionality reduction, variable reduction, hybrid networks, normal fuzzy and ANN have been applied to predict bond rating[11].

A recent online survey through the ISI Web of Knowledge using keywords such as "ANN" and "thermal design" would reveal only ten relevant SCI publications[12]. In the area of food processing, ANN was used to predict the maximum or minimum temperature reached in the sample after pressurization and the time needed for thermal re-equilibration[13]. The accurate determination of thermophysical properties of milk is very important for design, simulation, optimization, and control of food processing such as evaporation, heat exchanging, spray drying, and so forth. Generally, polynomial methods are used for prediction of these properties based on empirical correlation to experimental data. However, it was found that ANN presented a better prediction capability of specific heat, thermal conductivity, and density of milk than polynomial modeling and it was suggested as a reasonable alternative to empirical modeling for thermophysical properties of foods[14].

Numerical simulation of natural circulation boiling water reactor is important in order to study its performance for different designs and under various off-design conditions. It was found that very fast numerical simulations, useful for extensive parametric studies and for solving design optimization problems, can be achieved by using an ANN model of the system[15]. ANN models and GA were applied for developing prediction models and for optimization of constant temperature retort thermal processing of conduction heating foods[16]. ANN technique has been used as a new approach to determine the exergy losses of an ejector-absorption heat transformer (EAHT)[17]. The results show that the ANN approach has the advantages of computational speed, low cost for feasibility, rapid

turnaround, which is especially important during iterative design phases, and easy of design by operators with little technical experience.

Computational fluid dynamics approach is often employed for heat transfer analysis of a ball grid array(BGA) package that is widely used in the modern electronics industry. Owing to the complicated geometric configuration of the BGA package, an ANN was trained to establish the relationship between the geometry input and the thermal resistance output[18]. The results of this study provide the electronic packaging industry with a reliable and rapid method for heat dissipation design of BGA packages. Thermal spraying is a versatile technique of coating manufacturing implementing large variety of materials and processes. An ANN was developed to relate processing parameters to properties of alumina-titania ceramic coatings[19]. Predicted results show globally a well agreement with the experimental values.

It can be seen that applications of ANN in thermal design is scarce and this article aims to explore the application of an ANN in gas-fired cooktop burner design.

BACKGROUND

Cooktop Design Goals

Gases that trap heat in the atmosphere are often called greenhouse gases. They include carbon dioxide, nitrous oxide, methane, and ozone. Individuals can produce greenhouse gas emissions directly by burning oil or gas for home heating and cooking or indirectly by using electricity generated from fossil fuel burning. In the last 200 years, mankind has been releasing substantial quantities of greenhouse gases into the atmosphere. These extra emissions are increasing greenhouse gas(GHG) concentrations in the atmosphere, enhancing the natural greenhouse effect, which is believed to be causing global warming.

To combat the global warming problem, gas suppliers and manufacturers of cooking appliances are trying to find ways of improving energy efficiency with reducing greenhouse gas emissions. In view of the number of controllable factors and responses to be studied, Design of Experiments(DOE) is often used for such kind of empirical investigations. The authors therefore proposed to combine DOE technique with

the ANN approach for solving the multiple input and multiple output(MIMO) design problem. To achieve optimal thermal efficiency and greenhouse gas emissions, a back-propagation ANN was used to simulate the operating conditions and the implementation details are illustrated through a real-life case.

EMPIRICAL STUDY

A three factor, three level, Full Factorial Design(with 3 repetitions) was employed to investigate the complex relationships of three design parameters of a cooktop burner, viz. the Reynolds number, the equivalence ratio, and the load-height(distance from nozzle to bottom surface of cookware). The range of Reynolds number(Re) considered varies from 400 to 700. A tailor-made cooktop burner with a ring of 128 mm diameter is used and circular nozzles (diameter = 6 mm) were used in the experiment. Fuel-rich flames(corresponding to equivalence ratios ranging from 1.4 to 1.8) similar to real-life cooking situations were employed in the experiments. The load-height ranges from 24 mm to 32mm (corresponding to a H/d ratio varying from 4 to 8). To allow for different spacing between groups of nozzles, four configurations of the cooktop burner were considered, viz. 2-nozzle 6-section, 3-nozzle 4-section, 4-nozzel 3-section and 6-nozzel 2-section configurations. These nozzle configurations were labeled 1, 2, 3 and 4 respectively(Fig.2). A total of 108 experiments were carried out for each configuration. Experiment results showed that the configuration of 3-nozzle 4 section based on the predetermined input conditions of Re=550, EqR=1.6 and H/d=8 would give the best thermal efficiency(62%) and acceptable CO and NOx emissions,

- Burner efficiency model

The thermal efficiency of a burner is defined as the percentage of the thermal input transferred to the water in the loading vessel. It was determined by measuring the elapsed time for a standard 4 kg load of water to be heated from 30°C to 80°C and the corresponding consumption of LPG. Mathematically, the thermal efficiency is calculated as:

$$\eta = (MCp\Delta T/QH_v)*100\% \qquad (1)$$

where, M(kg) is the load mass of water, Cp(kJ/kg˚C) is the specific heat of water, Q(m³) is the LPG consumption, ΔT(˚C) is the temperature rise and H_v(kJ/m³) denotes the heating value of LPG.. An analysis of variance (ANOVA) on the data collected indicated that the quadratic regression model shown in Eqn.(2) could adequately describe the thermal efficiency (η). Stepwise method was used to remove the insignificant terms.

$$\eta = 0.49 - 0.05A - 0.038B + 0.17C - 0.039CD + 0.021B^2 - 0.18C^2 \tag{2}$$

Where, $-1 \leq A, B, C \leq 1$ corresponding to

$400 \leq Re \leq 700; 1.4 \leq Re \leq 1.8; 4 \leq H/d \leq 8$

The adjusted multiple correlation and the adequate precision coefficients were found to be 0.95 and 29.63 respectively.

• Burner emission models

Similarly, based on the values of Adjusted Multiple Correlation and Adequate Precision coefficients generated from the ANOVA results, the regression models shown in Eqn.(3) and Eqn.(4) adequately describe the CO and NOx emissions.

$$CO = 1790.22 + 745.27A + 557.01B - 1721.41C + 280.57AB - 251.97AC - 39.19BC + 31.54A^2 - 67.06B^2 + 481.87C^2 \tag{3}$$

$$NOx = 51.10 - 6.99A - 6.26B + 23.90C + 2.22AB - 3.77AC - 5.58BC \tag{4}$$

It is noted from the efficiency and emission models that the combined effects of some design factors can have considerable impact on the responses.

ANN OPTIMIZATION OF COOKTOP DESIGN

For this MIMO system, the architecture of the neural network can have several layers. Each layer has a weight matrix **W,** a bias vector **b**, and output vector **a**. Generally, a network of two layers, where the first layer is sigmoid and the second layer is linear, can be employed to approximate any function reasonably well

and such structure was adopted in this case. The two layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors.

The output of ith neuron in the hidden layer is:

$$a1_i = f_1(\Sigma wl_{ij}p_j + bl_i), i = 1,2; j = 1,2$$

For the output of kth neuron in the output layer is:

$$a2_k = f_2(\Sigma w2_{kj}al_i + b2_k), k = 1,2; j = 1,2$$

The error function is defined as:

$$E(W, B) = (1/2)\Sigma(t_k - a2_k)^2$$

Gradient Method was used for calculation of weight variation and the back-propagation of the error, when training the network.

Weight Variation of the output model is:

$$\Delta w2_{kj} = -\eta(\delta E/\delta w2_{kj}) = -\eta(\delta E/\delta a2_k)(\delta a2_k/\delta w2_{kj})$$

$$\Delta b2_{kj} = -\eta(\delta E/\delta b2_{kj}) = -\eta(\delta E/\delta a2_k)(\delta a2_k/\delta b2_{kj})$$

Weight variation of the hidden layer is:

$$\Delta w1_{kj} = -\eta(\delta E/\delta w1_{kj}) = -\eta(\delta E/\delta a2_k)(\delta a2_k/\delta w2_{kj})(\delta a1_i/\delta w1_{ij})$$

$$\Delta b1_i = \eta\delta_{ij}$$

Before training, it is necessary to scale the inputs and targets so that they always fall within a specified range. In the neural network model above, it is found that if preprocess and postprocess procedures are omitted, the network can hardly achieve the designed goal. This is due to large magnitude differences between the input and output parameters, e.g.: efficiency is in the range of 0.4 to 0.68, but CO emission(in ppm) is in the range of hundreds. All the input and output data was so scaled that they would fall within the range of [-1, 1]. The initialization of the weights and bias for each layer is important before training a feed-forward network. If improper values were chosen, training time would be excessive and the network could not convergent. As the characteristics of the actual operating condition are unknown, random initialization is used.

Figure 1. MSE trend
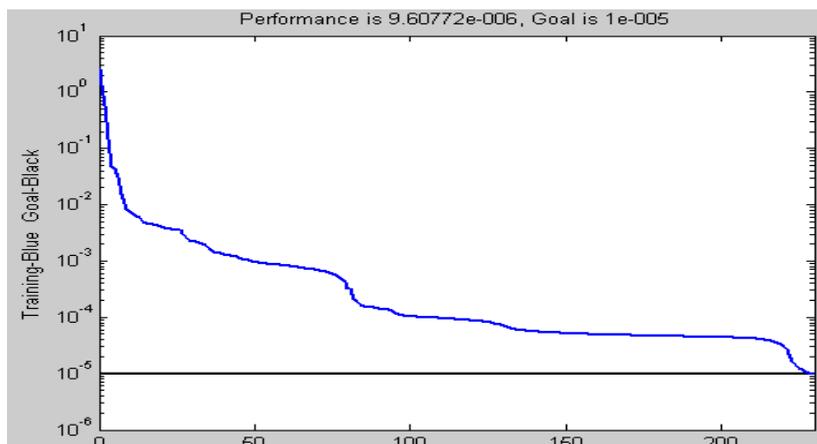
For each cooktop burner configuration, 90 sets of experiment results were used to train the neural network, and the remaining 18 were used to evaluate whether the network can represent the model of cooktop well. Mean square error(MSE) was used to test the performance of the network. It was found that the neural network performed reasonably well for the testing data. 32 neurons were used in the hidden layer, and for an average training time of 200~300 epochs, MSE could easily dropped to 10^{-5} (Fig. 1).

As the NN model performed quite well after training by using the actual experimental data, it is used to simulate the real environment of the experiments. After a simulation of all the reasonable combinations of the three initial factors, a maximum value of the thermal efficiency (corresponding to low CO and NOx emissions) was found.

There were four nozzle configurations in the simulation, the same as in the experiment. For each nozzle configuration, the number of simulations was:

$$[(700-400)/5]*[(1.8-1.4)/0.05]*[(8-4)/1] = 1,920$$

After the simulation, the first step was to choose the results that could meet the Chinese National Standard on Gas Appliances: thermal efficiency $\geq 60\%$, CO emission ≤ 300(ppm), NOx emission ≤ 100(ppm). The second step was to find the maximum value of the thermal efficiency, from this value the corresponding combination of input factors and configuration type of

cooktop were found. The simulation results showed that while the 2-nozzle and 3-nozzle configurations satisfied the National Standards, the 3-nozzle configuration indicates slightly better results: the thermal efficiency reaches 62.8% and the CO and NOx emissions are 257 and 91 respectively. Fig. 2 shows a comparison of thermal efficiency, CO and NOx emissions for each nozzle configuration respectively.

To confirm the validity of the NN results, three additional experiments based on the optimum values were carried out. Table 1 shows a comparison of the predicted and observed responses for the 3-nozzle burner configuration under optimal conditions. It can be seen that the predicted values of efficiency and NO_x emission are close to the experimental values. The difference between the predicted and the observed values are about 15%.

FUTURE TRENDS

Through this study one can see that artificial neural networks, as in other real-life applications, offer cooktop designers a highly versatile new tool for the thermal design of gas-fired cooktop burners. The successful utilization of this procedure requires that the ANN be fully optimized. In this study, backpropagation, a local search algorithm, was used for optimization and it is likely that local optimum rather than the global was obtained. One may attempt to address this problem

Figure 2. Comparison of ANN simulation results
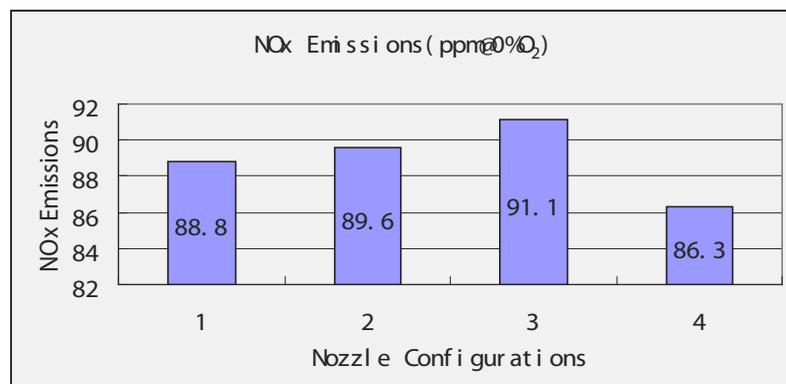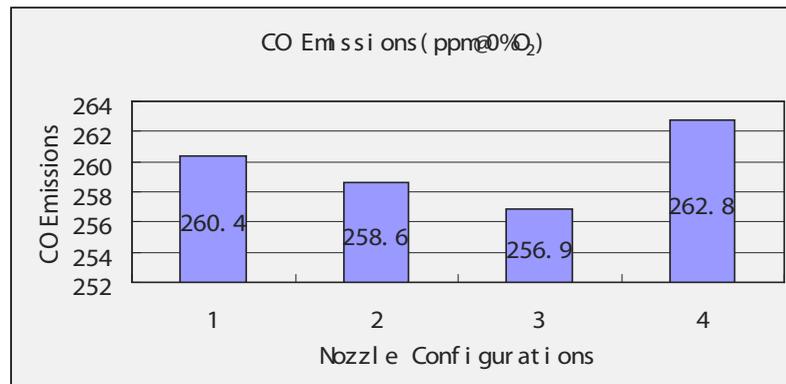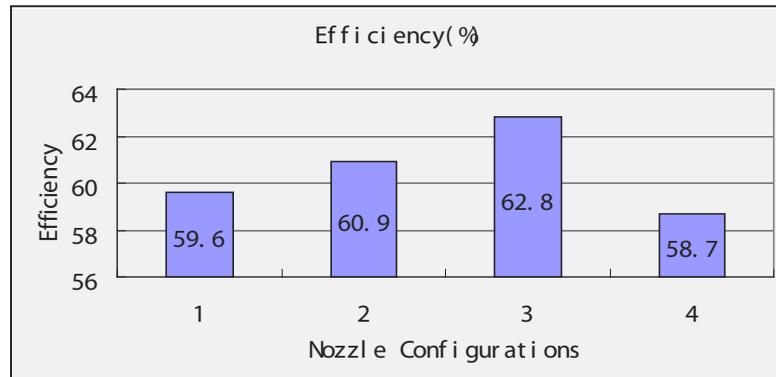
with *ad hoc* procedures such as stopping optimization at sub-optimal solutions(not over-training) or adjusting the neural network architecture to make optimization easier. As shown in other engineering applications[1], a promising alternative would be the use of a global optimization algorithm such as the genetic algorithm.

It is anticipated that by using a global search algorithm, the objective function can be set to balance the tradeoff between the over parameterization of the model that may over fit the data and a parsimonious ANN that can provide a more robust solution.

T

Table 1. A comparison of NN prediction and experimental results at optimal design

Re	EqR	H/d	Responses					
			Efficiency		CO (ppm@0%O_2)		NOx (ppm@0%O_2)	
			Predicted	Observed	Predicted	Observed	Predicted	Observed
580	1.65	8	0.63	0.60	257	292	91	98

CONCLUSION

A problem facing the gas-fired cooktop burner design community is the determination of design parameters which will result in a product with the most desirable combination of functional outcomes. Through an empirical investigation on the simultaneous optimization of thermal efficiency and GHG emissions, ANN methodology has proved to be an effective empirical modeling tool. Both the multiple correlation (R^2) and the mean square error (MSE) of ANN models for efficiency, CO and NO_x emissions indicated that the ANN results were quite satisfactory. Through multiple regression modeling one can evaluate the significance of the main and combined effects of various design parameters on burner efficiency and emissions. The relationship between the gas-fired burner design parameters and performance is hence further understood. To enhance the optimizing capabilities of the proposed ANN, a global search algorithm such as GA should be used instead of the backpropagation algorithm. It is believed that a GA-based ANN would provide a practical tool for the mechanical engineering design community.

ACKNOWLEDGMENT

REFERENCES

[1] Rabuñal, J.R. & Dorado, J. (2006). Artificial Neural Networks in Real-Life Applications, Hershey: Idea Group Publishing.

[2] Dorado, J., Pedreira, N. & Miguelez, M. (2006) Development of ANN with Adaptive Connections by CE. In Rabuñal, J.R. & Dorado, J. (Eds), Artificial Neural Networks in Real-Life Applications(pp.71-93), Hershey: Idea Group Publishing.

[3] Cortez, P., Rocha, M. & Neves, J. (2006). Time Series Forecasting by Evolutionary Neural Networks. In Rabuñal, J.R. & Dorado, J. (Eds), Artificial Neural Networks in Real-Life Applications(pp.47-70), Hershey: Idea Group Publishing.

[4] Rabuñal, J.R., Dorado, J., Pazos, A. Pereira, J. & Rivero, D. (2004). A new approach to the extraction of ANN rules and to their generalization capacity through GP. Neural Computation, 7(16), 1483-1523,

[5] Manrique, D., Rios, J. & Rodriguez-Paton, A. (2006) Self-Adapting Intelligent Neural S y s t e m s Using Evolutionary Techniques. In Rabuñal, J.R. & Dorado, J. (Eds), Artificial Neural Networks in Real-Life Applications(pp.94-115), Hershey: Idea Group Publishing.

[6] Pose, M.G., Carollo, A.C., Garda, J.M.A. & Gomez-Carraced, M.P. (2006). Several Approaches to Variable Selection by Means of Genetic Algorithms. In Rabuñal, J.R. & Dorado, J. (Eds), Artificial Neural Networks in Real-Life Applications(pp.141-165),Hershey: Idea Group Publishing.

[7] Loke, E., Warnaar, E.A., Jacobsen, P. Nelen, F. & Do Ceu Almedia, M. (1997). Artificial neural networks as a tool in urban storm drainage. Water Science and Technology, 36(8-9), 101-109.

[8] Dorado, J., Rabuñal, J.R., Pazos, A., Rivero, D., Santos, A. & Puertas, J. (2003). Prediction and modeling of the rainfall-runoff transformation of a typical urban basin using ANN and GP. Jr. of Applied Artificial Intelligence, 17, 329-343.

[9]   Gonzalez, B., Martinex, M. & Carro, D. (2006) Prediction of the Consistency of Concrete by Means of the Use of Artificial Neural Networks. In Rabuñal, J.R. & Dorado, J. (Eds), Artificial Neural Networks in Real-Life Applications(pp.188-200), Hershey: Idea Group Publishing.

[10]   Castillo, E., Cobo. A., Gutierrrez, J.M. & Pruneda, R.E. (1999). Massachusetts: Kluwer Academic Publishers.

[11]   Sethuraman, J. (2006) Soft Computing Approach for Bond Rating Prediction. In Rabuñal, J.R. & Dorado, J. (Eds), Artificial Neural Networks in Real-Life Applications (pp.202-219), Hershey: Idea Group Publishing.

[12] ISI Web of Knowledge. Retrieved on September 18, 2007 from http://portal.isiknowledge.com/portal.cgi/wos?Init=Yes&SID=4B@gEggbKJO5kMP27fn

[13] Torrecilla J.S., Otero L. & Sanz, P.D. (2007) Optimization of an artificial neural network for thermal/pressure food processing: Evaluation of training algorithms, Computers And Electronics in Agriculture, 56(2), 101-110.

[14] Mattar, H.L., Minim, L.A., Coimbra, J.S.R.(2004) Modeling thermal conductivity, specific heat, and density of milk: A neural network approach. International Jr. of Food Properties, 7(3), 531-539.

[15]Garg A., Sastry, PS, Pandey M, et al. (2007) Numerical simulation and artificial neural network modeling of natural circulation boiling water reactor, Nuclear Engineering and Design, 237(3), 230-239.

[16]Chen, C.R., Ramaswamy, H.S. (2002) Modeling and optimization of constant retort temperature (CRT) thermal processing using coupled neural networks and genetic algorithms, Jr. of Food Process Engineering, 25(5), 351-379.

[17]Sozen A. & Arcaklioglu, E. (2007) Exergy analysis of an ejector-absorption heat transformer using artificial neural network approach, Applied Thermal Engineering, 27 (2-3), 481-491.

[18]Ho, C.I., Hung, T.C. & Hung, C.I. (2005) Thermal analysis and optimization for a ball grid array package, Proc. of Inst. Mech. Engrs, Pt C - Jr of Mechanical Engineering Science, 219 (4), 381-393.

[19]Guessasma, S., Montavon, G.. & Coddet, C. (2004) Plasma spray process modelling using artificial neural networks: Application to Al2O3-TiO2 (13% by weight) ceramic coating structure, Jr. de Physique, IV(120), 363-370.

KEY TERMS

Backpropagation: A supervised learning technique used for training artificial neural networks. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). The term is an abbreviation for "backwards propagation of errors". Backpropagation requires that the transfer function used by the artificial neurons (or "nodes") be differentiable.

Full Factorial Design: This design allows a designer to adequately quantify a response with a reasonable number of tests. In general, full factorial designs require three levels for each factor thus allowing one to evaluate second order models.

Designed of Experiments: A set of tests conducted under controlled conditions in which multiple levels of a set of factors are manipulated and the resulting response(s) of a system or process is measured or observed.

Factors: The set of (independent) variables that are believed to affect the response of a system or process.

Levels: The sets of values for each factor.

Multilayer Perceptrons (MLPs): They are feed-forward neural networks trained with the standard backpropagation algorithm. They are supervised networks so they require a desired response to be trained. They learn how to transform input data into a desired response, so they are widely used for pattern classification. With one or two hidden layers, they can approximate virtually any input-output map.

Multiple Correlation, R^2: the percent of variance in the dependent variable explained collectively by all of the independent variables.

Response: The (dependent) variable(s) measured or observed that is the result of a test conducted at a specific set of factor levels.

T

Reynolds Number(Re): the ratio of inertial forces to viscous forces and consequently it quantifies the relative importance of these two types of forces for given flow conditions. Thus, it is used to identify different flow regimes, such as laminar or turbulent flow.

A 2D Positioning Application in PET Using ANNs

Fernando Mateo
Universidad Politécnica de Valencia, Spain

Ramón J. Aliaga
Universidad Politécnica de Valencia, Spain

Jorge D. Martínez
Universidad Politécnica de Valencia, Spain

José Mª Monzó
Universidad Politécnica de Valencia, Spain

Rafael Gadea
Universidad Politécnica de Valencia, Spain

INTRODUCTION

Positron Emission Tomography (PET) is a radiotracer imaging technique based on the administration (typically by injection) of compounds labelled with positron emitting radionuclides to a patient under study. When the radio-isotope decays, it emits a positron, which travels a short distance before annihilating with an electron. This annihilation produces two high-energy (511 keV) gamma photons propagating in nearly opposite directions, along an imaginary line called Line of Response (LOR).

In PET imaging, the photons emitted by the decaying isotope are detected with gamma cameras. These cameras consist of a lead collimator to ensure that all detected photons are propagated along parallel paths, a crystal scintillator to convert high-energy photons to visible light, photo-multiplier tubes (PMT) to transform light signals into electric signals, and associated electronics to determine the position of each incident photon from the light distribution in the crystal (Ollinger & Fessler, 1997).

We have researched on how Artificial Neural Networks (henceforth ANNs or NNs) could be used for bias-corrected position estimation. Small-scale ANNs like the ones considered in this work can be easily implemented in hardware, due to their highly parallelizable structure. Therefore, we have tried to take advantage of the capabilities of ANNs for modelling the real detector response.

BACKGROUND

Traditionally, Anger logic (Anger, 1958) has been the most popular technique to obtain the the position of the *centroid*, or centre of the light distribution inside the scintillator crystal by means of a simple formula. The solution proposed by Anger involves connecting the PMT outputs to a simple resistor division circuit to obtain only four signals (X⁻, X⁺, Y⁻, Y⁺). However, Anger logic introduces some important drawbacks in the detection process: non-uniform spatial behaviour, differences between each PMT gain or the deformation of the light distribution when it approaches the edge of the scintillator. These problems are alleviated by using correction maps.

However, the presence of all these phenomena in traditional detectors still reduces the intrinsic resolution and produces non-uniform compression artifacts in the image and the so called border effects. The main consequence is an unavoidable reduction of the *Useful Field Of View* (UFOV) of the PET camera, which usually covers up to 60% of each crystal dimension.

With other methods such as *Statistics Based Positioning* (SBP) or *Maximum Likelihood* (ML) positioning, this UFOV can be increased to approximately the 80% of each dimension of the crystal, but these methods involve a heavier computational cost (Joung, Miyaoka, Kohlmyer & Lewellen 2001)(Chung, Choi, Song, Jung, Cho, Choe, Lee, Kim & Kim, 2004).

These drawbacks have not been fully overcome yet. Therefore, our proposal to introduce ANNs in the detection process as good quality estimators is well-grounded.

Some previous research has been made in this area for PMT (A.M. Bronstein, M.M. Bronstein, Zibulevsky & Zeevi, 2003) and Avalanche Photodiode (APD) based (Bruyndockx, Léonard, Tavernier, Lemaître & Devroede, 2004) detectors using neural networks. In this work, the detectors are based on continuous scintillators and Multi-Anode PMTs (MA-PMTs) employing charge division read-out circuits (Siegel, Silverman, Shao & Cherry, 1996).

ANN APPROACH TO 2D POSITIONING IN PET

Materials and Methods

We have employed the GEANT4 (Agostinelli, 2002) simulation toolkit to model the detector and to generate realistic inputs for the NN. The electronic read-out of the resistor circuit was performed with SPICE analysis. The supervised training and validation of the ANNs have been carried out with the MATLAB Neural Networks

Figure 1. Siegel's DPC diagram

Toolbox (The Mathworks, Inc., 2004). We have chosen the RPROP algorithm (Riedmiller & Braun, 1993) because it proved to converge faster than the standard gradient descent algorithm and other variants such as the Levenberg-Marquardt algorithm. Radial basis (RB) networks were also considered but were discarded in the end due to their inferior performance.

Detector Characteristics

The model of the detector under study comprises a $49 \times 49 \times 10$ mm^3 continuous slab of LSO scintillator crystal coupled to a Hamamatsu H8500 Flat-Panel MA-PMT. The read-out electronics is a conventional DPC-like resistive charge division circuit that proves to model Anger's logic accurately. Taking the resistor network pattern used by Aliaga et al. (Aliaga, Martínez, Gadea, Sebastiá, Benlloch, Sánchez, Pavón & Lerche, 2006) as a starting point, we have designed a new resistor network based on the architecture proposed by S. Siegel (Siegel, Silverman, Shao & Cherry, 1996) (Fig. 1) that allows us to estimate the 2D positioning with better results. As in the previous design, all 64 channels (one per anode of the H8500) are coded into only 4 output lines, which are then fed into current sensitive preamplifiers. The current-ratio matrices A, B, C and D corresponding to each output were obtained from electronic read-out using SPICE analysis. The network was analyzed applying the superposition theorem for electric circuits.

Neural Networks

Given a collimated source *S* of γ photons with origin at (x_s, y_s, z_s) emitting perpendicularly to the detector surface, we can describe the interaction of a photon in the detector as a random variable $X \rightarrow \mathbf{A}$, being, $\mathbf{A}$ a vector of elements a_i, the number of photoelectrons arriving at each anode of the MA-PMT. Thus, the elements of the vector $\mathbf{J}$ are the inputs of the NN, which can be written as

$$J_k = \sum_i A_i \cdot G_i \cdot R_{i,k} \qquad (1)$$

where J_k is the *k*th output of the charge division network, *G* the vector of pad gains of the MA-PMT (in our case randomly distributed between 1 and 3) and

1577

$R_{i,k}$ the transfer function of the DPC from the ith anode to the kth output of the resistors network.

The Universal Approximation Theorem (Haykin, 1999) claims that any continuous function, defined over a determined region, can be approximated uniformly, with arbitrary precision, by a Multi-Layer Perceptron (MLP) of two hidden layers. Then, our position estimator can be expressed as

$$\hat{\mathbf{r}} = \Phi\{\mathbf{J}; W, b\}$$

being W and b the weights and biases of each neuron of the NN. In order to adapt the NN estimator to a function f, we begin from a training set composed of pairs (J_i, X_i) where $X_i = (x_i, y_i)$ is the position of the source and $J_i = f(X_i)$ is a realization of the outputs of the charge division network for an interaction with origin at position X_i. Thus, the weights and biases of the NN are modified following a gradient descent algorithm (*backpropagation*) to minimize the mean squared error

$$E = \frac{1}{2}\sum_i \left\| X_i - F\left(J_i\right)\right\|^2$$

where F is the transfer function of the NN. Initial values of weights and biases are usually determined following the Nguyen-Widrow rule (Nguyen & Widrow, 1990).

Optimization

The detector surface was partitioned in 49×49 positions of 1 mm² each. An amount of 1000 valid events were generated on each position using GEANT4. Of these 1000 events, 500 were used to compose training subsets and the remaining 500 to compose test subsets, as depicted in Figure 2.

For each network architecture, 20 different trainings were averaged, each one with different initial weights and arbitrary gains for each anode. The number of epochs was fixed to 800 to ensure convergence in all cases. We preferred not to use cross-validation as there was no scarcity of patterns to train the network. The chosen activation function was the hyperbolic tangent (*tanh*).

We reduced our study to MLPs of two hidden layers. A third layer would only increase complexity without showing any significant improvement. Our analysis showed that an increment on the number of neurons in the second hidden layer improves the linearity of the response, reducing the systematic error, while an increase in the number of neurons in the first hidden layer improves the spatial resolution.

There are two different approaches for 2D positioning: a single NN with 4 inputs and 2 outputs, or two independent NNs for 1D positioning on each axis. For the first scheme, we have simulated MLPs with two

Figure 2. Methodology to obtain the training/test subsets
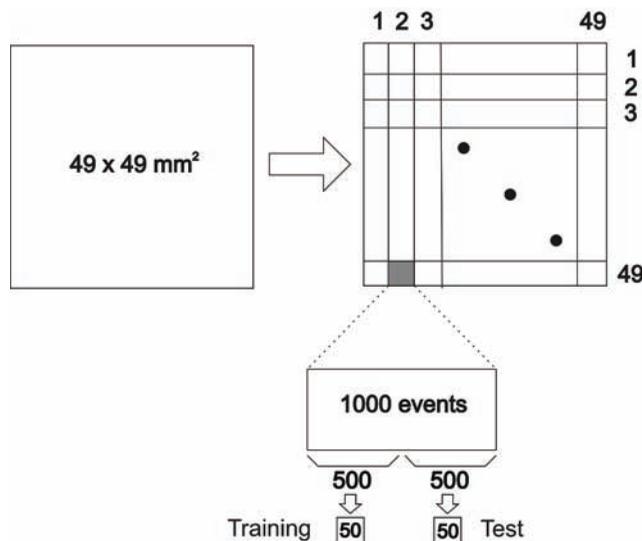
Figure 3. 2D positioning histogram for (a) centroid (Anger) based estimator and (b) NN estimator, using a 49x49 grid with 1 mm spacing

T

(a)
(b)
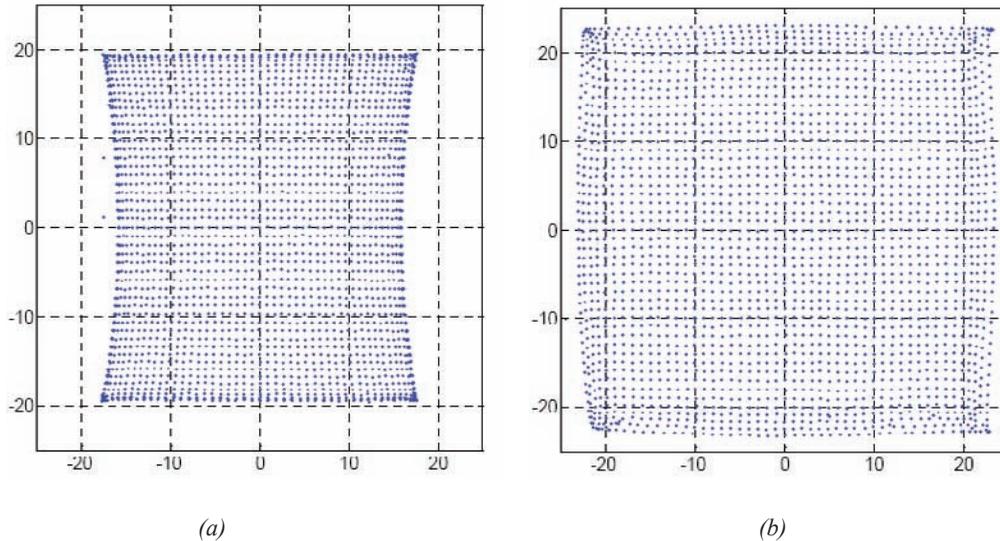
hidden layers and up to 25 hidden neurons (considering $N_1 + N_2 \leq 25$, and $N_1 > N_2$, where N_1 is the number of neurons in the first hidden layer and N_2 is the number of neurons in the second one. Thus, the NN architectures are represented as: number of inputs/N_1/N_2/number of outputs) to prevent overtraining.

Results

The optimum network architecture found using a single MLP was 4/15/8/2. However, the best results were achieved with the double MLP estimator, with a 4/9/6/1 architecture, reaching a mean systematic error below 0.4 mm at almost all the detector FOV.

In Fig. 3, we can observe a 2D positioning histogram for a grid of 49×49 points separated 1 mm both using a centroid estimator (a) and a NN estimator (b).

The figure clearly shows that the centroid approach introduces non-uniform compression artifacts in the borders of the crystal, while the NN estimator successfully corrects these artifacts and produces a more uniform FOV in both dimensions. This improvement is quite significant as the UFOV increases from de 30×30 mm² to 40×40 mm², which means approximately a 90% of the MA-PMT effective area for normal incidence (Mateo, Aliaga, Martínez, Monzó & Gadea, 2007).

FUTURE TRENDS

It would be desirable to develop a method to extend this approach to Depth Of Interaction (DOI) estimation, especially to deal with oblique incidence. With this objective, additional work is being carried out by our group (Lerche, Benlloch, Sánchez, Pavón, Giménez, Fernández, Giménez, Escat, Cerdá, Martínez &. Sebastiá, 2005). This would add a fifth input, Z, to the ANN, which would enable a very accurate and fully 3D reconstruction of the interaction point within the scintillator.

It would also be interesting to implement the ANN training on a hardware platform, to perform fast on-site trainings to and to enable us to calibrate the PET instrumentation automatically.

And last, but not least, we are working on a high precision testbench (Fig. 4), which has recently been presented in the Real Time Conference 2007 (Monzó, Aliaga, Herrero, Martínez, Mateo, Sebastiá, Mora, Benlloch & Pavón, 2007), allows to link several simulation tools for each part of the PET system, enabling us to model the effects of the electronics in each part of the design separately. In this setup, there are separate analog and digital parts. The analog part is composed by an Application Specific Integrated Circuit (ASIC),

Figure 4. Block diagram of the high precision testbench under development
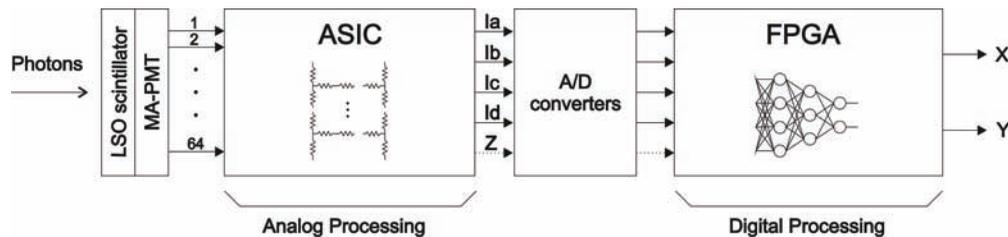
and it includes an implementation of our DPC. The digital part consists of a Field Programmable Gate Array (FPGA) board, where the neural network is going to be embedded, among other elements related to signal processing. In addition to that, we intend to install a radioactive source to obtain real stimuli instead of our current synthetic data.

CONCLUSION

ANNs have proved to be good position estimators for PET, and an interesting alternative to traditional Anger logic. The benefits of using ANN-based position estimators include lower systematic errors, and also lower standard deviations of the systematic error, increased UFOV (up to 90% of the MA-PMT effective area, for normal incidence), less compression artifacts on the crystal borders and slightly better spatial resolution, especially on the borders.

Regarding the DPC circuit, it allowed a reduction of complexity both in terms of number of variables and in terms of hardware resources.

REFERENCES

Agostinelli, S. (2002). *GEANT4: A Simulation Toolkit*. Stanford Linear Accelerator Center, Stanford University, Stanford, CA.

Aliaga, R.J., Martínez, J.D., Gadea, R., Sebastiá, A., Benlloch, J.M., Sánchez, F., Pavón, N., & Lerche, Ch. (2006). Corrected position estimation in PET detector modules with multi-anode PMTs using neural networks. *IEEE Transactions on Nuclear Science,* 53(3), 776 – 783.

Anger, H. (1958). Scintillation camera, *Review of Scientific Instruments,* 29(1), 27–33.

Bronstein, A.M., Bronstein, M.M., Zibulevsky, M., & Zeevi, Y.Y. (2003). Optimal nonlinear line-of-flight estimation in positron emission tomography. *IEEE Transactions on Nuclear Science*, 50(3), 421–426.

Bruyndockx, P., Léonard, S., Tavernier, S., Lemaître, C., Devroede, O., Wu, Y., & Kreiguer, M. (2004). Neural network-based position estimators for PET detectors. *IEEE Transactions on Nuclear Science*, 51(5), 2520–2525.

Chung, Y.H., Choi, Y., Song, T.Y., Jung, J.H., Cho, G., Choe, Y.S., Lee, K.-H., Kim, S.E., & Kim, B.-T. (2004). Evaluation of Maximum-Likelihood Position Estimation With Poisson and Gaussian Noise Models in a Small Gamma Camera. *IEEE Transactions on Nuclear Science*, 51(1), 101-104.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

Joung, J., Miyaoka, R.S., Kohlmyer, S.G., & Lewellen, T.K. (2001). Investigation of Bias-Free Positioning Estimators for the Scintillation Cameras. *IEEE Transactions on Nuclear Science*, 48(3), 715-719.

Lerche, Ch.W., Benlloch, J.M., Sánchez, F., Pavón, N., Giménez, N., Fernández, M., Giménez, M., Escat, B., Cerdá, J., Martínez, J.D., & Sebastiá, A. (2005). Depth of gamma-ray interaction within continuous crystals from the width of its scintillation light-distribution. *IEEE Transactions on Nuclear Science,* 52(3), 560–572.

Mateo, F., Aliaga, R.J., Martínez, J.D., Monzó, J.M., & Gadea, R. (2007). Incidence Position Estimation in a PET Detector Using a Discretized Positioning Circuit

Useful Field of View (UFOV): Area of the scintillator crystal surface on which the incidence of gamma rays produces reasonable estimations of the position of incidence.

2D–PAGE Analysis Using Evolutionary Computation

Pablo Mesejo
University of A Coroña, Spain

Enrique Fernández-Blanco
University of A Coroña, Spain

Diego Martínez-Feijóo
University of A Coroña, Spain

Francisco J. Blanco
Juan Canalejo Hospital, Spain

INTRODUCTION

This paper presents the preliminary studies for the creation of a new tool to assist in medical diagnostic. The tool will help in the analysis of 2D-PAGE images. In order to create a 2D-PAGE image of an ideal patient—the patient could be healthy or ill—the tool will help us in the creation of an image that facilitates and speeds up future diagnostics. The creation of a master image has motivated the development of a tool to alignment gel images. The tool will make easier the correspondence among the proteins into the ideal image and the ones of a new image. Due to the fact that image registering process is quite complex, we use the Intel's library OpenCV which provides functions to calculate optical flow and translation vectors.

This library introduces into the project a set of variables unknown by the facultative. To solve this, an automatic selection of values for this set of variables is necessary. This last task is made with the Evolutionary Computation technique called Particle Swarm Optimization (Kennedy, R. & Eberhart, J. 1995)

BACKGROUND

In the 20th century medicine, the number of medical images has been growing. X-ray photographies, magnetic resonances, 2D gels images, angiographies can be taken as examples. The major difficulty for the physician is to integrate all this information in order to offer a diagnosis.

This way, since computers started being used for analyzing and treating images at the end of the 20th century, one of the most important fields inside the application of computers to image processing has been the treatment of all medical existing images. It is here where technologies of Evolutionary Computation and Neural Networks are necessary, because they facilitate certain processes of adjustment that, in another way, would be extraordinarily complex or laborious. Among the most usual technologies used for the processing of biomedical images there can be pointed out Artificial Neural Networks, Genetic Algorithms, Particle Swarm Optimization, Splines or Growth of Regions.

Amongst some examples, we can emphasize the use of Artificial Neural Networks for the analysis of radiological images, Genetic Algorithms (Holland, J.H.,1975) in the 3D reconstruction of anthropologic models (Santamaría, J., Cordón, O., Damas, S., Alemán, I., Botella, M., 2006) and in the integration of the information obtained by means of different methods—Computed Tomography (CT), Magnetic Resonance Imaging (MRI),…—(Rouet, J. M., Jacq. J. J., Roux, C., 2000), Particle Swarm Optimization for alignment of 2D and 3D biomedical images (Wachowiak, M. P., Smolikova, R., Zheng, Y., Zurada, J. M., Elmaghraby, A. S., 2004) or the use of Splines to 2D-PAGE registering (Seow, N., Sowmya, A., Sun, C., 2005).

In our case, the technology to use will be the Particle Swarm Optimization dedicated to improve the analysis of 2D-PAGE (Seoane, J. A., Mesejo, P., Ruiz-Romero, C., Dorado, J., Pazos, A., Blanco, F. J., 2007).

PSO APPLIED TO THE OPTIMIZATION OF OPENCV PARAMETERS FOR 2D-PAGE ANALYSIS

The aim of this investigation is to help the doctors in process of identification of certain characteristics in the 2D-PAGE images (Ruiz-Romero, C., López-Armada, M. J., Blanco, F. J., 2005). In order to do that the registry image process will consist on the alignment between the master image, which has been labelled for every protein, and an image whose interesting points have been identified by the facultative to study. This registry process will make easier to the medical the study of the presence or absence of a certain kind of protein and its concentration.

2D-PAGE

This work uses the images called 2D-PAGE—*poly-acrylamide gel electrophoresis*. The process to obtain these images uses the electrophoresis, which is a well known analytic technique for macromolecules—DNA or Protein—separation. The responsible of this separation is the mobility presented by the electrically charged macromolecules when a differential voltage is applied. The method tries to immobilize the studying molecules into a gelatinous material; in this case the material will be polyacrylamide. This process was well described by (Bueno García, G. 2005) as "A differential voltage is applied to the gel with the biological samples inside during a concrete period of time. Each molecule will migrate through the gel pores with a different speed, which is dependent of the electrical charge and the mass of each molecule."

On one hand, the resultant gels are classified on X-axis on respect the isoelectric point—PH to which an amphoteric substance has no voltage. On the other hand, on Y-axis gels are sorted by their molecular mass.

The resultant image will help us to detect the presence or absence of a certain protein, or even the more or less protein concentration. This information will assist us to know the existence or inexistence of some illness or characteristic.

OpenCV

OpenCV is an open source library which has been developed in C++ for Computer Vision. This library is optimized to be applied in real time problems inde-pendently of the platform. It is especially oriented to images manipulation and processing and also to the movement analysis into the image. Some interesting information about this library could be found at (Agam, G., 2006).

To solve the problem, one of the Optical Flow functions, which can be found in the OpenCV library, was used. Specifically we use the function known as *CalcOpticalFlowBM*. This function divides two images into blocks, in order to find the same block of the first one in the second image. After this search process, the function establishes a set of movement vectors which corresponds to the movements of the blocks of the image. For more information about this topic visit (Department of Electrical Engineering, Nara National College of Technology's Web Page, 2006) and (Intel's Web Page).

This function is very useful in our problem because we need an alignment image tool for the creation of the proteomic diagnostic image. When we try to align two images, a function that compares these two images and tells us the movement among them will be useful. This function searches the similarities using the statistical correlation among sets of pixels of the two images. The new protein location process will only need the movement vector and the block of that spot.

The function *CalcOpticalFlowBM* has a set of parameters, the ones to be optimized are:

- **blockSize:** the size of comparable blocks in which the image is divided.
- **maxRange:** neighborhood maximum size around a block that would be explore to find the block in the second image.

These parameters will be optimized with an artificial intelligence technique, because, in other case, this process will be done manually by the user.

Particle Swarm Optimization

The Evolutionary Computation technique that has been used in this work is the Particle Swarm Optimization (PSO). Inspired on the social swarm from nature, was developed by Kennedy and Eberhart (Kennedy, J. & Eberhart, R., 1995). In a PSO algorithm a particle swarm explores the search space. Each particle represents a possible solution to the optimization problem. The position of each particle is the result of the best position

visited by the particle—self experience—and the best position of its neighborhood—its social experience. When the particle's neighborhood is the whole swarm, the best particle in the neighborhood is the best global position; the resulted algorithm is called *gBest* PSO. When smaller neighborhood size is used, the algorithm is called *lBest* PSO. The fitness of each particle—how far from the optimum is—is calculated using a function that varies with the concrete optimization problem.

Each particle is represented into swarm by:

- x_i: particle's current position.
- v_i: particle's current speed.
- y_i: particle's best self-position

Particle´s best self-position for an i element is the best position visited by the particle. If f is the objective function then the best self-position in a time t is updated as (Ec.1):

$$y_i(t+1) = \begin{cases} y_i(t), & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t), & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases}$$

$$(1)$$

If the particle's best global position is denoted by the vector $\hat{y}$ then:

$$\hat{y}(t) \in \{y_0, y_1, ..., y_s\} = \min\{f(y_0(t)), f(y_1(t)), ..., f(y_s(t))\}$$

$$(2)$$

where s denotes the swam size. Taking *lBest* with N_j neighborhoods in which the best neighborhood's particle is denoted by $\hat{y}_j$, that particle will be known as the best neighborhood's particle and is defined by:

$$\hat{y}_j(t+1) \in \{N_j \mid f(\hat{y}_j(t+1)) = \min\{f(y_i(t))\}, \forall y_i \in N_j\}$$

$$(3)$$

where

$$N_j = \{y_{i-l}(t), y_{i-l+1}(t), ..., y_{i-1}(t), y_i(t), y_{i+1}(t), ..., y_{i+l-1}(t), y_{i+l}(t)\}$$

$$(4)$$

The neighborhoods are usually defined by the particle's index, but can be also defined by topological relations. It is easy to see that *gBest* is only a particular case of *lBest*, where the neighborhood is the whole swarm. Notice that the *lBest* produces more diversity in the solutions, but it is also true that it has a higher computation time cost than *gBest*.

For each PSO algorithm iteration, the update of speed v_i is specified for each dimension $j \in 1, ..., N_d$, being N_d the dimension of the problem. So, v_{ij} represents the jth element of the ith particle's speed vector. Burning on mind this, the i particle speed is updated by the following equation:

$$v_{ij}(t+1) = wv_{ij} + c_1 r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(\hat{y}_j(t) - x_{ij}(t))$$

$$(5)$$

the most important terms are:

- *Learning rates (weights), c_1 and c_2,* which determine the influence of the learning components, cognitive versus social.

- $r_{1j}, r_{2j} \sim U(1,0)$, these components introduce the randomness into the algorithm.
- The term inertia component, w, is used to control the influence of the previous speed. High values of this term increase the global exploration; however, low values increase the local exploitation.

- The cognitive component, $y_i(t) - x_i(t)$, represents the particle's self-experience to find the better solution.

- The social component, $\hat{y}(t) - x_i(t)$, represents the swarm knowledge about the better solution.

The position of particle i, x_i, is updated using the equation 6:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

$$(6)$$

The PSO algorithm applies repeatedly the update equations, which have been described previously. Until a number of iterations is exceeded, the update speeds are near 0. The fitness value of the function gives us the quality of the solution.

In concrete the PSO algorithm that has been used to optimize the OpenCV parameters is a complete model. The values of c_1 and c_2 are not zero and then the social and cognitive components are taken into account. Finally, to comment that these values were fixed to the test in the value of 2 as is recommended in (Kennedy, J. and Mendes, R., 2006). The inertia weights are fixed between the values 0,8 and 1,2 (Eberhart, R. and Shi, Y., 2000).

For more information about PSO the following papers could be consulted: (Omran, M.G., Engelbrecht, A.P. and Salman A., 2005) and (Kennedy, J., Eberhart, R. C., and Shi, Y., 2001).

Combining all Elements

The PSO algorithm is used, in this case, to establish the block size and the search area in the images. To do that we try to minimize the intensity differences of each pixel for every couple of blocks considerate as equals.

When the PSO will have ended we will have the optimus parameter configuration to the OpenCV function.

Some examples of this application are in Figure 1.

In the previous left figure we can see the image of a new gel to analyze, from which interests to us the three red remarked proteins. After the application of block matching technique and optimizing the OpenCV parameters with the PSO system, we obtain the correct position of the proteins on the master image, as we can see in the central image.

FUTURE TRENDS

After the correct location of the protein, by means of the already seen technologies, the main future aim would consist on the integration of the intensity of the protein in the master image. A possible way of integrating the new proteins in the master would be to use the alpha channel (transparency of the image) so that the fragment of the image to integrate had the values corresponding to the alpha level to one and those of the target image to zero, thereby this fragment would join the target image. With the intention of the integration process being as smooth as possible in the edges of spot integrated; one degraded will be realized by means of a low pass filter. The low pass filter emphasizes the Low Frequencies, smoothes images and noises, reducing the variability of the image. A median filter could be used to make this smoothing, this filter is less sensible to extremely asides values. Also the degraded could be made manually, choosing a size of window that includes the interest point, so that the degraded between the margin of the window and the limit of the point was proportional to the distance to the spot.

The final aim would be, by means of some technology of artificial intelligence (Artificial Neural Networks or Expert Systems), to advise the doctor in the diagnosis, for which, obviously, the previous tool should have been calibrated and adjusted correctly.

Figure 1. (Left) New gel; (center) master; (right) new gel with the intermediately step
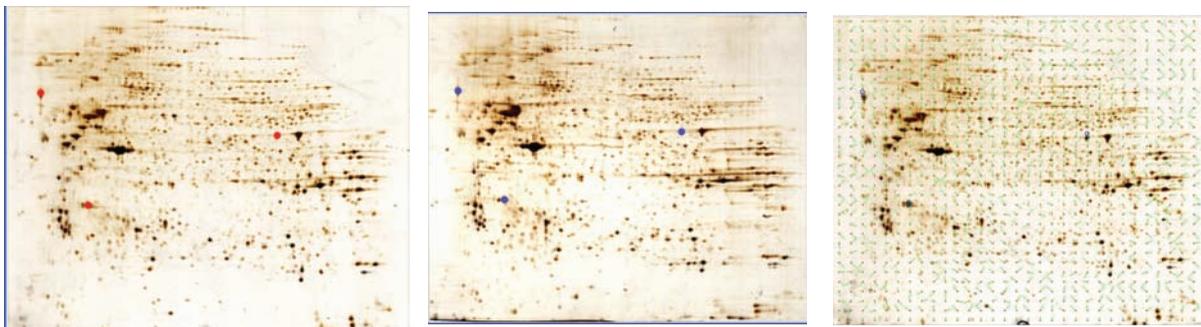
CONCLUSION

The first thing that we can extract after this work is that technologies of Evolutionary Computation can be used to assist in medical decision. This way, it has been proved that by means of these technologies, a support has been realized in the identification of certain interesting points. This is useful for the doctors since they will not have to know the functioning of the tool of computation, in particular the parameters that control the function of the library that executes the alignment.

Besides, these technologies of computation do not obtain a unique adjustment but a set of these, which will allow to choose the best result from the point of view of the user.

REFERENCES

Agam, G. (2006). http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/index.html

Bueno García, G. (2005) *Procesado de Imagen Molecular Genómica e Histológica*. In *Tic en biomedicina.* Rabuñal, J. R., Gestal, M., Pedreira, N., Pereira, J.(eds)

Darwin, D. (1859) *On the origin of species by means of natural selection*.

Department of Electrical Engineering, Nara Nathional College of Technology (2006) http://robotics.elec.nara-k.ac.jp/opencv/ref/OpenCVRef_Motion_Tracking.htm#ch4_optflow

Eberhart, R. and Shi, Y. (2000) *Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization*. In *Proceedings of the International Congress on Evolutionary Computation*, vol. 1, pp. 84-88.

Fogel, L.J., Owens, A.J. & Walsh, M.A. (1966) *Artificial Intelligence through Simulated Evolution*. Wiley, NY.

Goldberg, D.E. (1989) *Genetics Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems.* University of Michigan Press. Ann Arbor.

Intel's Web Page about OpenCV, http://www.intel.com/technology/computing/opencv/index.htm

Kennedy, J. and Eberhart, R. (1995). *Particle Swarm Optimization, Proc. 1995 IEEE Intl. Conf. on Neural Networks, pp. 1942-1948,* IEEE Press**.**

Kennedy, J., Eberhart, R. C., and Shi, Y., (2001) *Swarm intelligence*. San Francisco Morgan Kaufmann Publishers.

Kennedy, J. and Mendes, R. (2006) *Neighborhood Topologies in Fully Informed and Best-of-Neighborhood Particle Swarms.* Man and Cybernetics, Part C, IEEE Transactions on Systems.

Omran, M.G., Engelbrecht, A.P. and Salman A., (2005), *Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification,* on *Pattern Analysis & Applications* Springer, London.

Rouet, J. M., Jacq. J. J., Roux, C., (2000) *Genetic Algorithms for a Robust 3-D MR-CT Registration*. IEEE Transactions on Information Technology in Biomedicine, VOL. 4, No. 2.

Ruiz-Romero, C., López-Armada, M. J., Blanco, F. J., (2005). *Proteomic characterization of human normal articular chondorcytes: A novel tool for the study of osteoarthritis and other rheumatic diseases*. Proteomics, pp 3048-3059

Santamaría, J., Cordón, O., Damas, S., Alemán, I., Botella, M., (2006) *3D Forensic Model Reconstruction by Scatter Search-based Pair-wise Image Registration*. Pp. 6006-6012, IEEE Press

Seoane, J. A., Mesejo, P., Ruiz-Romero, C., Dorado, J., Pazos, A., Blanco, F. J. (2007) *Diagnóstico por imagen en reumatología: de la imagen radiológica a la imagen molecular.* In I+S: Informática y salud, 62: pp. 9-17.

Seow, N., Sowmya, A., Sun, C., (2005) *Multi-image 2D-PAGE Feature Detection*. Proceedings of the Digital Imaging Computing: Techniques and Applications.

Wachowiak, M. P., Smolíková, R., Zheng, Y., Zurada, J. M., Elmaghraby, A. S., (2004) *An Approach to Multimodal Biomedical Image Registration Utilizing Particle Swarm Optimization*. IEEE Transactions On Evolutionary Computation, 8, 3, 2004, pp. 289-301, IEEE Press.

T

Wallace, A. R. (1858) *On the Tendency of Varieties to Depart Indefinitely From the Original Type*

KEY TERMS

Amphoteric Substance: Substance is one that can react as either an acid or base.

Area of the Search Space: Set of specific ranges or values of the input variables that constitute a subset of the search space.

Artificial Neural Networks: System composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.

Electrophoresis: Separation of molecules (proteins or nucleic acids) in an electric field as a function of their molecular weight and/or their electric charge.

Evolutionary Computation: Generic term used to indicate any population-based metaheuristic optimization algorithm that uses mechanisms inspired by biological evolution (Darwin, D., 1859) (Wallace, A. R., 1858), such as reproduction, mutation and recombination.

Genetic Algorithm: An algorithm for optimizing a property based on an evolutionary mechanism that uses replication, deletion, and mutation processes carried out over many generations. (Goldberg, D.E., 1989) (Fogel, L.J., Owens, A.J. & Walsh, M.A. 1966)

Particle: Each of the elements that explore the search space in a Particle Swarm Optimization algorithm.

Particle Swarm Optimization: Evolutionary Computation technique that basis its functioning on natural swarm behaviour like the birds. This algorithm uses a swarm of particles to explore the search space

Polyacrylamide: Acrylate polymer formed from acrylamide subunits that is readily cross-linked.

Protein: A molecule composed of a long chain of amino acids. Proteins are the principal constituents of cellular material

Search Space: Set of all possible situations of the problem that we want to solve could ever be in.

Visualizing Cancer Databases Using Hybrid Spaces

Julio J. Valdés
National Research Council Canada, Canada

Alan J. Barton
National Research Council Canada, Canada

INTRODUCTION

According to the World Health Organization(WHO), the directing and coordinating authority for health within the United Nations system *http://www.who.int/cancer/en/*, from a total of 58 million deaths in 2005, cancer accounts for 7.6 million (or 13%) of all deaths worldwide. This places cancer as one of the leading causes of death in the world, with lung cancer (the main cancer leading to mortality) accounting for 1.3 million deaths per year. Thus the importance of understanding the mechanisms of lung cancer is clear. One approach is through the rapid quantification of the gene expression levels of samples of healthy and diseased lung tissue. This new field blending the knowledge from biologists, computer scientists and mathematicians is known as Bioinformatics and is yielding large quantities of data of a very high dimensional nature that needs to be understood.

BACKGROUND

The increasing complexity of the data analysis procedures makes it more difficult for the user (not necessarily a mathematician or data mining expert), to extract useful information out of the results generated by the various techniques. This makes graphical representation directly appealing; for which Virtual Reality (VR) is a suitable paradigm. Virtual Reality is *flexible*; it allows the construction of different virtual worlds representing *the same* underlying information, but with a different look and feel. VR allows *immersion*, that is, the user can navigate inside the data, interact with the objects in the world. VR creates a *living* experience. The user is not merely a passive observer but an actor in the world. VR is *broad and deep*. The user may see the VR world as a whole, and/or concentrate the focus of attention on

specific details of the world. Of no less importance is the fact that in order to interact with a Virtual World, no mathematical knowledge is required, and the user only needs minimal computer skills. A virtual reality technique for visual data mining on heterogeneous, imprecise and incomplete information systems was introduced in (Valdés, J.J., 2002) (Valdés, J.J., 2003) (see also *http://www.hybridstrategies.com*).

The purpose of this article is to explore the construction of high quality VR spaces for visual data mining (in opposition to classical data mining (Fayyad, U., Piatesky-Shapiro, G., & Smyth, P., 1996)) using a multi-objective optimization technique applied to the understanding of a publicly available lung cancer gene expression data set. This approach provides both a solution for the previously discussed problem, and the possibility of obtaining a set of spaces in which the different objectives are expressed in different degrees, with the proviso that no other spaces could improve any of the considered criteria individually (if spaces are constructed using the solutions along the Pareto front). This strategy represents a conceptual improvement in comparison with spaces computed from the solutions obtained by single-objective optimization algorithms in which the objective function is a weighted composition involving different criteria.

THE MULTI-OBJECTIVE APPROACH: A HYBRID PERSPECTIVE

In order to establish a formulation of the problem based on multi-objective optimization, a set of objective functions has to be specified, representing the corresponding criteria that must be simultaneously satisfied by the solution. The minimization of a measure of similarity information loss between the original and the transformed spaces and a classification error measure

over the objects in the new space can be used in a first approximation. Clearly, more requirements can be imposed on the solution by adding the corresponding objective functions. Following a principle of parsimony this paper will consider the use of only two criteria, namely, Sammon's error (Sammon, J.W., 1969) for the unsupervised case and mean cross-validated classification error with a k-nearest neighbour pattern recognizer for the supervised case.

The proximity (or similarity) of an object to another object may be defined by a distance (or similarity) calculated over the independent variables and can be defined by using a variety of measures. In the present case a normalized Euclidean distance is chosen:

$$d_{\overline{x}\, \overline{t}} = \sqrt{(1/p) \sum_{j=1}^{p} (x_{ij} - t_{kj})^2} \tag{1}$$

Structure Preservation: An Unsupervised Perspective

Examples of error measures frequently used for structure preservation (Kruskal, J., 1964) (Sammon, J.W., 1969) (Borg, I., & Lingoes, J., 1987) are:

$$\text{S stress} = \sqrt{\frac{\sum_{i<j} (\delta_{ij}^2 - \zeta_{ij}^2)^2}{\sum_{i<j} \delta_{ij}^4}}, \tag{2}$$

$$\text{Sammon error} = \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(\delta_{ij} - \zeta_{ij})^2}{\delta_{ij}} \tag{3}$$

$$\text{Quadratic Loss} = \sum_{i<j} (\delta_{ij} - \zeta_{ij})^2 \tag{4}$$

For heterogeneous data involving mixtures of nominal and ratio variables, the Gower similarity measure (Gower, J.C., 1973) has proven to be suitable. The similarity between objects *i* and *j* is given by

$$S_{ij} = \sum_{k=1}^{p} s_{ijk} / \sum_{k=1}^{p} w_{ijk} \tag{5}$$

where the weight of the attribute (w_{ijk}) is set equal to 0 or 1 depending on whether the comparison is consid-

ered valid for attribute k. If $v_{k(i)}$, $v_{k(j)}$ are the values of attribute k for objects i and j respectively, an invalid comparison occurs when at least one them is missing. In this situation w_{ijk} is set to 0.

For quantitative attributes (like the ones of the datasets used in the paper), the scores s_{ijk} are assigned as

$$s_{ijk} = 1 - |v_k(i) - v_k(j)|/R_k$$

where R_k is the range of attribute k. For nominal attributes

$$s_{ijk} = \begin{cases} 1 & \text{if } v_k(i) = v_k(j) \\ 0 & \text{otherwise} \end{cases}$$

This measure can be easily extended for ordinal, interval, and other kind of variables. Also, weighting schemes can be incorporated for considering differential importance of the descriptor variables.

Multi-Objective Optimization Using Genetic Algorithms

An enhancement to the traditional evolutionary algorithm (Bäck T., Fogel, D.B., & Michalewicz, Z, 1997), is to allow an individual to have more than one measure of fitness within a population. One way in which such an enhancement may be applied, is through the use of, for example, a weighted sum of more than one fitness value (Burke, E.K., & Kendall, G., 2005). Multi-objective optimization, however, offers another possible way for enabling such an enhancement. In the latter case, the problem arises for the evolutionary algorithm to select individuals for inclusion in the next population, because a set of individuals contained in one population exhibits a Pareto Front (Pareto, V., 1896) of best current individuals, rather than a single best individual. Most (Burke, E.K., & Kendall, G., 2005) multi-objective algorithms use the concept of dominance to address this issue.

A solution $x_{(1)}$ is said to dominate (Burke, E.K., & Kendall, G., 2005) a solution $x_{(2)}$ for a set of m objective functions <$f_1(x)$, $f_2(x)$, ..., $f_m(x)$> if

- $x_{(1)}$ is not worse than $x_{(2)}$ over all objectives. For example, $f_3(x_{(1)}) \le f_3(x_{(2)})$ if $f_3(x)$ is a minimization objective.

V

- $x_{(1)}$ is strictly better than $x_{(2)}$ in at least one objective. For example, $f_6(x_{(1)}) > f_6(x_{(2)})$ if $f_6(x)$ is a maximization objective.

One particular algorithm for multi-objective optimization is the elitist non-dominated sorting genetic algorithm (NSGA-II) (Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T., 2000), (Deb, K., Agarwal, S., Pratap, A., & Meyarivan, T., 2000), (Deb, K., Agarwal, S., & Meyarivan, T., 2002), (Burke, E.K., & Kendall, G., 2005). It has the features that it *i)* uses elitism, *ii)* uses an explicit diversity preserving mechanism, and *iii)* emphasizes the non-dominated solutions.

Original Study

Gene expressions were compared in (Spira, A., Beane, J., Pinto-Plata, V., Kadar, A., Liu, G., Shah, V., Celli, B., & Brody, J.S., 2004) for severely emphysematous lung tissue (from smokers at lung volume reduction surgery) and normal or mildly emphysematous lung tissue (from smokers undergoing resection of pulmonary nodules). The original database contained 30 samples (18 severe emphysema, 12 mild or no emphysema), with 22,283 attributes. Genes with large detection P-values were filtered out, leading to a data set with 9,336 genes that were used for subsequent analysis. Nine classification algorithms were used to identify a group of genes whose expression in the lung distinguished severe emphysema from mild or no emphysema. First,

model selection was performed for every algorithm by leave-one-out cross-validation, and the gene list corresponding to the best model was saved. The genes reported by at least four classification algorithms (102 genes) were chosen for further analysis. With these genes, a two-dimensional hierarchical clustering using Pearson's correlation was performed that distinguished between severe emphysema and mild or no emphysema. Other genes were also identified that may be causally involved in the pathogenesis of the emphysema. Data was from: *http://www.ncbi.nlm.nih.gov/projects/geo/gds/gds_browse.cgi?gds=737*.

Experimental Settings

Each sample in this study is a vector in a high dimensional space, and therefore, direct inspection of the structure of this data, and of the relationship between the descriptor variables (the genes) and the type of sample (normal or cancer), is impossible. Moreover, within the collection of genes there is a mixture of potentially relevant genes with others which are irrelevant, noisy, etc. The need of simultaneously finding a visual representation (3D) respecting (as much as possible) the set of object interrelationships as defined by the original attributes, and the construction of a new feature space effectively differentiating the two classes of objects present, makes this problem suitable for a multi-objective optimization approach.

Table 1. Experimental settings for computing the pareto-optimal solution approximations by the multi-objective genetic algorithm (PGAPack (Levine, D., 1996) extended by NSGA-II).

population size	100	number of generations	500
chromosome length	$90 (= 3 \cdot 30)$	ga seed	101
No. new inds. in $(i + 1\text{st})$ pop.	20	objective functions should be	minimized
chromosome data representation	real	crossover probability	0.8
crossover type	uniform (prob. 0.6)	mutation probability	0.4
mutation type	gaussian	selection type	tournament
tournament probability	0.6	mutation and crossover	yes
population initialization	random, bounded	lower bound for initialization	-2
upper bound for initialization	2	fitness values	raw
stopping criteria	maximum iterations	restart ga during execution	no
parallel populations	no		
number of objectives	2	number of constraints	0
pre-computed diss. matrix	Gower dissimilarity		
evaluation functions	mean cross-validated k-nn error and Sammon error		
cross-validation (c.v.)	5 folds	randomize before c.v.	yes
knn seed	101	k nearest neighbors	3
non-linear mapping measure	Sammon	dimension of the new space	3

The collection of parameters describing the application of the NSGA-II algorithm is shown in Table-1. A modest population size and number of generations were used, with a relatively high mutation probability in order to enable richer genetic diversity. Randomization of the set of data objects was applied in order to reduce the bias in the composition of the cross-validated folds by providing a more even class distribution between successive training and test subsets. The number of folds was set in consideration of the sample size.

Results

The set of non-dominated solutions obtained by the NSGA-II algorithm is shown in the scatter plot of Fig-1(a), where the horizontal axis is the mean cross-validated knn error and the vertical axis the Sammon error. The approximate location of the Pareto front is defined by the convex polygon joining the solutions provided by chromosomes 2, 1, 10, etc. Chromosome 2 defines a space with a perfect resolution of the supervised problem in terms of the "*no or mild emphysema*" and "*severe emphysema*" classes (knn error =0), but at the cost of a severe distortion of the space. Whereas, chromosome 1 approximates a pure unsupervised solution (with low Sammon error). Its classification error is large indicating that few non-linear features preserving the similarity structure lacks classification power. This may be due to the large amount of attribute noise, redundancy, and irrelevancy within the set of 22,283 original genes.

Clearly, it is impossible to represent virtual reality spaces on a static medium. However, a composition of snapshots of the VR spaces using the solutions along the Pareto front approximation is shown in Fig-1(b-d). Different mappings (even with important differences from the point of view of the mapping error) lead to similar 3D visual representations, which indicate good solution reproducibility. The similarities are associated to the main distributions of the clouds of points, which are preserved, while there might be local discrepancies with respect to the placement of some objects.

A solution satisfying classification error as much as possible (actually with 0-error) is shown in Fig-1(b) where both classes are separated into 2 main clouds of points and a distinct point, Object 6, positioned separately from the clouds. It can be seen that Object 6 is positioned relatively differently in the spaces that comprise the best Sammon error Fig-1(d) and trade-offs

between the classification error objective and Sammon error objective Fig-1(c). This is why, visually, the latter space represents a compromised solution between the two goals and why it is a trade-off between the two objective functions. It should be remembered that the class information is not used at all for computing the spaces. Chromosome 10, according to Fig-1(a) and Fig-1(c), can be considered to be the best multi-objective compromised solution in which both error criteria are simultaneously as low as possible. It shows reasonable class discrimination with a non-large similarity structure distortion, which is a very meaningful result.

FUTURE TRENDS

Visualization of data is of potential interest for various research communities and the authors have applied various visualization approaches to other medical data diseases such as those coming from Scleroderma Skin disease, Breast Cancer, Alzheimer's disease, and Leukemia. But, in fact, a restriction to medical data is not made by the authors, for which they have also preliminarily investigated data coming from, for example, the fields of Hydrochemistry and Geophysical Prospecting.

CONCLUSION

A multi-objective optimization approach was introduced for the problem of computing virtual reality spaces in the context of visual data mining and knowledge discovery applied to relational structures (e.g. databases). The multi-objective procedure was based on NSGA-II using two objective functions representative of unsupervised and supervised criteria (mean cross-validated knn error as a measure of miss-classification, and Sammon error as a measure of similarity structure loss). This methodology was applied to the analysis of high dimensional genomic data collected in the framework of Lung cancer research. A Pareto front approximation was recognizable from within the solutions provided by the final population. Selected solutions from along that approximation were used for the construction of a sequence of visualizations showing the progression from spaces with complete class separation and poor similarity preservation to spaces with reversed characteristics. A solution with a reasonable compromise

Figure 1. Set of 100 multi-objective solutions. Those along the Pareto front approximation progressively span the extremes between minimum classification error and minimum dissimilarity loss. 3 solutions were selected and snapshots of VR spaces computed. Geometries: ``light grey spheres'' = no or mild emphysema samples, ``dark grey spheres encased within a convex hull'' = severe emphysema samples. Behavior = static.

V
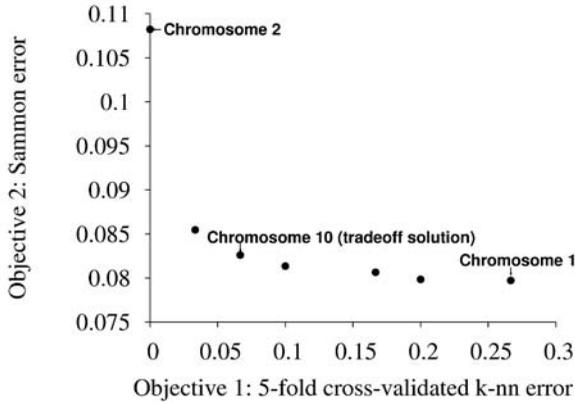
(a) Front obtained by multi-objective optimization (NSGA-II) that approximates the true Pareto Front.
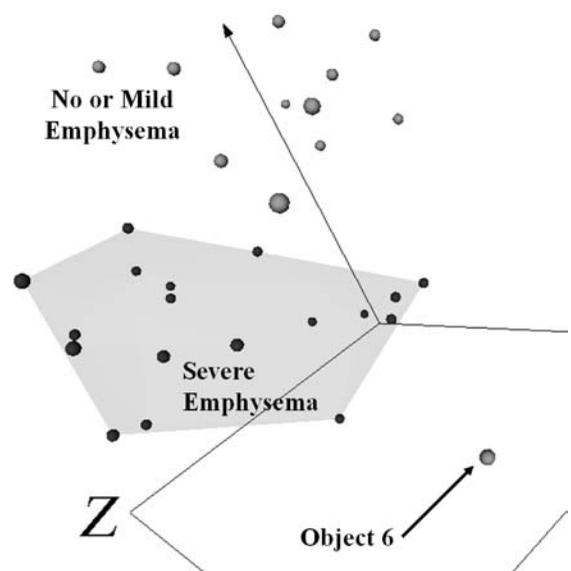
(b) Chromosome 2 (5-fold CV k-nn Error: 0.0000, Sammon Error: 0.1082)
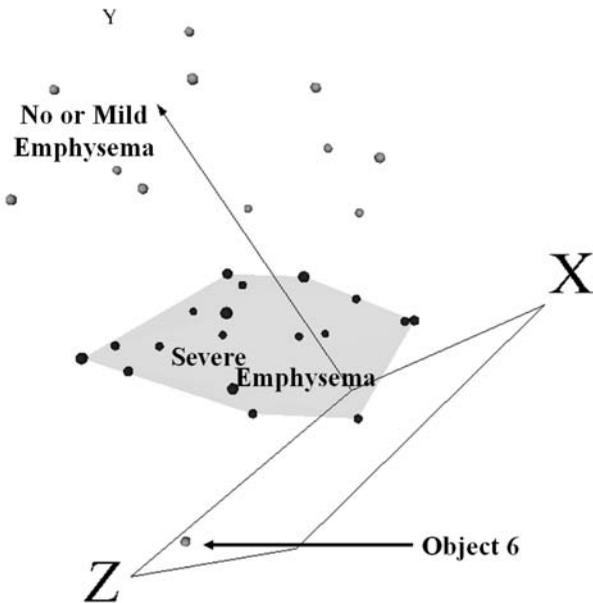
(c) Chromosome 10 (5-fold CV k-nn Error: 0.0667, Sammon Error: 0.0826)
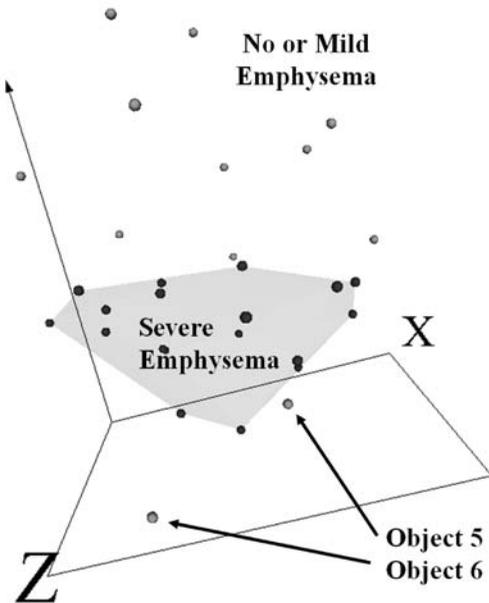
(d) Chromosome 1 (5-fold CV k-nn Error: 0.2667, Sammon Error: 0.0797)

1593

between the two criteria was identified and clearly contained properties of both extreme solution spaces. These research results, although preliminary, showed large potential and further investigation is required.

ACKNOWLEDGMENT

REFERENCES

Bäck T., Fogel, D.B., & Michalewicz, Z (1997). *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford Univiversity Press.

Borg, I., & Lingoes, J. (1987). *Multidimensional similarity structure analysis*. Springer-Verlag.

Burke, E.K., & Kendall, G. (2005). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer Science and Business Media, Incorporated.

Deb, K., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6 (2), 181-197.

Deb, K., Agarwal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, 849-858.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2000). A fast and elitist multi-objective genetic algorithm: Nsga-ii. *Technical Report 2000001, Kanpur Genetic Algorithms Laboratory*, Indian Institute of Technology Kanpur.

Fayyad, U., Piatesky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery. In U.F. et al., editor, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 1-34.

Gower, J.C. (1973). A general coefficient of similarity and some of its properties. *Biometrics*, 1(27):857-871.

Kruskal, J. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psichometrika*, 29:1-27.

Levine, D. (1996). *Users Guide to the PGAPack Parallel Genetic Algorithm Library*. Argonne National Laboratory, Argonne, IL.

Pareto, V. (1896). *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne.

Sammon, J.W. (1969). A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, C18:401-408.

Spira, A., Beane, J., Pinto-Plata, V., Kadar, A., Liu, G., Shah, V., Celli, B., & Brody, J.S. (2004) Gene Expression Profiling of Human Lung Tissue from Smokers with Severe Emphysema. *American Journal of Respiratory Cell and Molecular Biology* 31, 601-610.

Valdés, J.J. (2002). Virtual reality representation of relational systems and decision rules. In P. Hajek, editor, *Theory and Application of Relational Structures as Knowledge Instruments*, Prague, Meeting of the COST Action 274.

Valdés, J.J. (2003). Virtual reality representation of information systems and decision rules. In *Lecture Notes in Artificial Intelligence*, LNAI 2639, Springer-Verlag, 615-618.

KEY TERMS

Cancer: A term for diseases in which abnormal cells divide without control and can invade other tissues. Cancer cells can spread to other parts of the body through the blood and lymph systems. Cancer is not just one disease but many diseases. There are more than 100 different types of cancer. http://www.cancer.gov/cancertopics/what-is-cancer

Evolutionary Algorithms: A subset of evolutionary computation, which generally only involve techniques inspired by biological evolution such as reproduction, mutation, recombination, natural selection and survival of the fittest. Candidate solutions to an optimization problem play the role of individuals in a population, and

the fitness function determines the environment within which the solutions "live". Evolution of the population then takes place after the repeated application of the above operators.http://en.wikipedia.org/wiki/Evolutionary_Computation

Gene: 1. A unit of DNA that carries information for the biosynthesis of a specific product in the cell. 2. Ultimate unit by which inheritable characteristics are transmitted to succeeding generations in all living organisms. Genes are contained by, and arranged along the length of, the chromosome. The gene is composed of deoxyribonucleic acid (DNA). Each chromosome of a species has a definite number and arrangement of genes, which govern both the structure and metabolic functions of the cells and thus of the entire organism. Genes provide information for the synthesis of enzymes and other proteins and specify when these substances are to be made. Alteration of either gene number or arrangement can result in mutation (a change in the inheritable traits).http://www.amfar.org/cgi-bin/iowa/bridge.html?page=G

Hybrid Space: A constructed space that attempts to preserve more than one property (possibly in conflict) of the original space. For example, preserving distances between objects and the class structure of the original space.

Multi-objective Algorithm: An optimization algorithm that attempts to find the best solutions across all measures of solution acceptability. That is, the Pareto Front is sought, even under the situation that it may not be theoretically known.

Unsupervised Algorithm: The true class that an object belongs to is not known to the algorithm; hence the algorithm is not supervised by a "teacher". For example, clustering algorithms are unsupervised because each cluster is generated based on the data itself. Although the true class may also be known, it was not used.

Virtual Reality: (often called VR for short) Is an attempt to provide more natural, human interfaces to software. It can be as simple as a pseudo 3D interface or as elaborate as an isolated room in which the computer can control the user's senses of vision, hearing, and even smell and touch. http://www.saugus.net/Computer/Terms/

NOTE

Voltage Instability Detection Using Neural Networks

Adnan Khashman
Near East University, Turkey

Kadri Buruncuk
Near East University, Turkey

Samir Jabr
Near East University, Turkey

INTRODUCTION

The explosive growth in decision-support systems over the past 30 years has yielded numerous "intelligent" systems that have often produced less-than-stellar results (Michalewicz Z. et al., 2005). The increasing trend in developing intelligent systems based on neural networks is attributed to their capability of learning nonlinear problems offline with selective training, which can lead to sufficiently accurate online response. Artificial neural networks have been used to solve many problems obtaining outstanding results in various application areas such as power systems. Power systems applications can benefit from such intelligent systems; particularly for voltage stabilization, where voltage instability in power distribution systems could lead to voltage collapse and thus power blackouts.

This article presents an intelligent system which detects voltage instability and classifies voltage output of an assumed power distribution system (PDS) as: stable, unstable or overload. The novelty of our work is the use of voltage output images as the input patterns to the neural network for training and generalizing purposes, thus providing a faster instability detection system that simulates a trained operator controlling and monitoring the 3-phase voltage output of the simulated PDS.

BACKGROUND

Artificial Neural Networks have been used to solve many problems obtaining outstanding results in various applications such as classification, clustering, pattern recognition and forecasting among many other applications corresponding to different areas.

Power system stability is the property of a power system which enables it to remain in a state of equilibrium under normal operating conditions and to regain an acceptable state of equilibrium after a disturbance. Beyond a certain level, the decrease of power system stability margins can lead to unacceptable operating conditions and/or to frequent power system collapses (Sjostrom M. et al., 1999) (Ernst D. et al., 2004). In 2003 and within less than two months, a number of blackouts happened around the world, affecting millions of people. These blackouts include (Novosel D. et al., 2004):

- The 14th of August blackout in Northeast United States and Canada, which is considered one of the worst blackouts in the history of these countries, affecting approximately 50 million people.
- The 28th of August blackout in London, which affected commuters during the rush hour.
- The 23rd of September blackout in Sweden and Denmark, which affected approximately 5 million people.
- The 28th of September blackout in Italy, which is considered the worst blackout in Europe ever, affecting approximately 57 million people.

In recent years voltage instability has been one of the major reasons for blackouts, and it is the root cause of the 14 August blackout. Voltage stability is threatened when a disturbance increases the reactive power demand beyond the sustainable capacity of the available reactive power resources. Although, progress in the areas of communication and digital technology has increased the amount of information available at the efficient supervisory control and data acquisition

(SCADA) systems, however, during events that cause outages, an operator may be overwhelmed by the excessive number of simultaneously operating alarms, which increases the time required for identifying the main outage cause and then starting the restoration process (De Souza A.C.Z. et al., 1997) (Lukomski R. & Wilkosz K., 2003). Additionally, factors such as stress and human error can affect the operator's performance; thus, the need for an additional tool to support the real-time decision-making process which currently exists. This tool can be in the form of an intelligent voltage instability detector.

The implementation of neural networks for stabilizing power systems in general has been recently suggested (Wenxin L. et al., 2003) (Cardoso G. et al., 2004) (Keyhani A. et al., 2005) (Alcántara F.J. & Salmerón P., 2005) (Mishra, 2006). Research on different approaches to the assessment and improvement of voltage stabilization in particular has proposed different solutions to voltage instability using neural networks (Bansilal et al., 2003) (Kamalasadan S., 2006) (Lin H.C., 2007). However, none of the existing intelligent system solutions to detecting voltage instability in power distribution systems addresses the possibility of providing an artificial intelligent detector that simulates a human operator whose task is to detect voltage instability via monitoring the voltage output.

This article suggests a novel method for detecting voltage instability in power distribution systems. The proposed system uses 3-phase voltage output images as its database for training and generalizing a supervised neural network based on the back propagation learning algorithm. The intelligent system comprises two phases: the image processing phase, where voltage output images are pre-processed and meaningful features are obtained as the input patterns for the next phase which is the neural network implementation. Here, the supervised neural network learns to associate the voltage output patterns with three possible classifications; namely, Stable, Unstable or Overload.

The main objective of the proposed intelligent system is to provide earlier detection of voltage instability thus aiding a human operator. The intelligent system can be operated concurrently with SCADA systems thus enhancing the stability of the power distribution system. Upon the detection of voltage instability by the intelligent system, further measures can be taken to quickly sustain stability or clean voltage drop of the power distribution system.

THE INTELLIGENT DETECTION SYSTEM

The intelligent voltage instability detection system comprises two phases. Firstly, the image processing phase; where the PDS voltage output graph images are processed and feature vectors are extracted to be used for training and/or testing the neural network. Secondly, the neural network implementation phase, where the extracted features from the first phase are used as input vectors to a neural network. Our neural network is based on the back propagation learning algorithm due to its implementation simplicity, and the availability of sufficient database for training this supervised learner.

Voltage Output Image Processing

Training and generalizing a neural network using images requires sufficient number of images and meaningful input patterns. Our database contains voltage output images that correspond to a MATLAB-simulated power system. Our concern is with the transient stability of one distribution power substation whose voltage readings are taken as outputs of the circuit after simulation for 20 seconds, which is considered sufficient time to assure the simulation of the three output cases; in particular the overload case. These outputs are graphs of the sinusoidal waves of voltage during the 20 seconds of simulation. For every second on the graph there are 50 full waves, which make them concentrated and appear like a block.

The intelligent voltage instability detection system has three possible output classifications (*Stable*, *Unstable* or *Overload*). The image database has to account for the three cases. For each case there are three voltage output graphs representing three voltage phases (a, b, c). A total number of 54 cases (18 stable, 18 unstable and 18 overload) are simulated, thus resulting in 162 voltage output graph images which form our database. Figure 1 shows examples of the image database representing the voltage output cases (stable, unstable and overload).

The objective of the image processing phase is the extraction of meaningful patterns which form the input to the neural network within the intelligent system. The extracted patterns should distinctly represent the different voltage output cases, while keeping their size to a minimum, in order to reduce the computational cost.

Figure 1. Voltage output image examples

| Stable | Unstable | Overload |
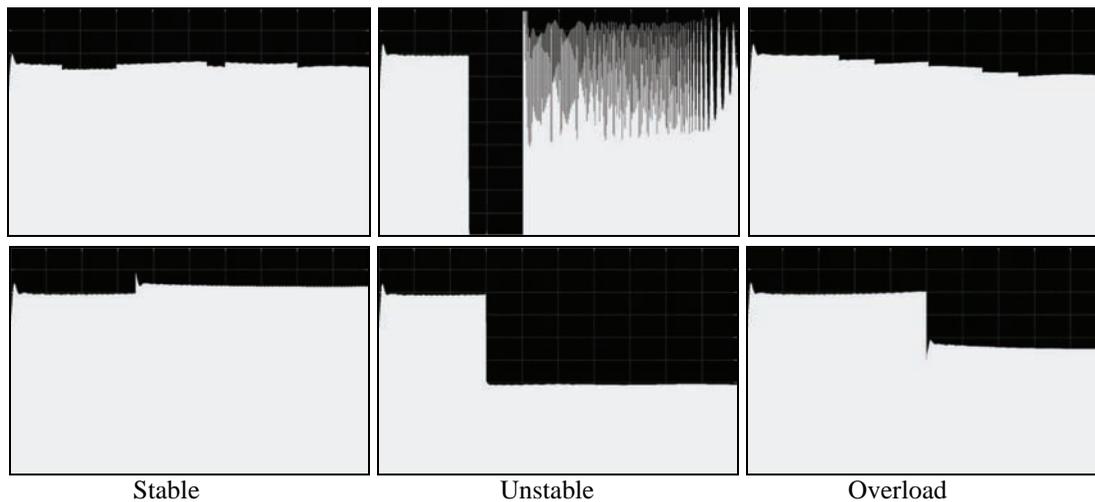
Figure 2 shows an example of finding pixel positions (row numbers) at which the highest voltage value is recorded. The patterns are extracted and saved as a feature vector using the following procedure:

- The three output voltage graphs (representing 3-phase voltage) for every case are saved as digital images with a size of (540x800) pixels.
- Every image is converted to grey and then resized to (202x400) pixels.
- In every image and for every column starting from column 2 to column 201 and from first row to last row, the value of the pixel where the first grey level discontinuity occurs is found and the number of this row is saved in a vector.
- This saved value represents the highest voltage value at that column in that image.
- The process of recording the row numbers where the highest voltage value occurs is repeated for the 200 columns, thus yielding a feature vector with 200 values for each voltage output graph.
- As a result, each case is represented by a pattern or feature vector with 600 values (200 values for each graph. 3 graphs representing 3-phase voltages for each case).

- The number of patterns is equal to the number of cases (54 patterns).
- The 600 values within each pattern are normalized to values from "0" to "1" using division by 400 which is the highest number of rows.
- The normalized patterns are then used as inputs to the neural network classifier for training or generalization.

Neural Network Topology

The second phase in our intelligent detection system is the implementation of the neural network which uses the patterns that were extracted from the voltage output image database. A total of 162 patterns each with 600 normalized values are available for this implementation. Training the neural network uses 30 cases (10 of each: stable, unstable and overload), thus, 90 patterns are used for training the neural network. Testing or generalizing the trained neural network uses the remaining 72 patterns that represent the other 24 cases (8 of each: stable, unstable and overload).

The neural network consists of an input layer with 600 neurons receiving the normalized values in each pattern, one hidden layer with 28 neurons which as-

Figure 2. Example on finding pixel positions at highest voltage values for a voltage unstable case. a- Resized grey image, b- Pixel positions of grey level discontinuities
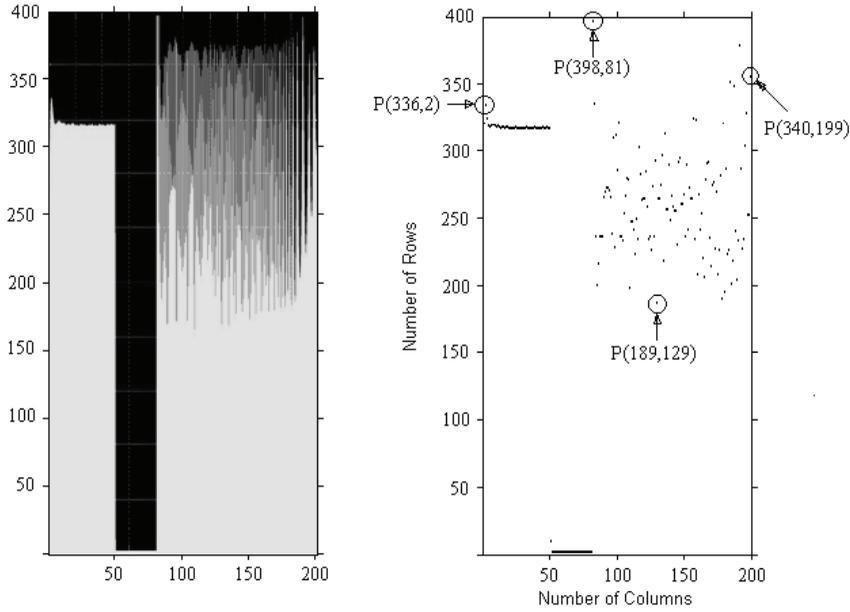
Figure 3. The intelligent voltage instability detection system
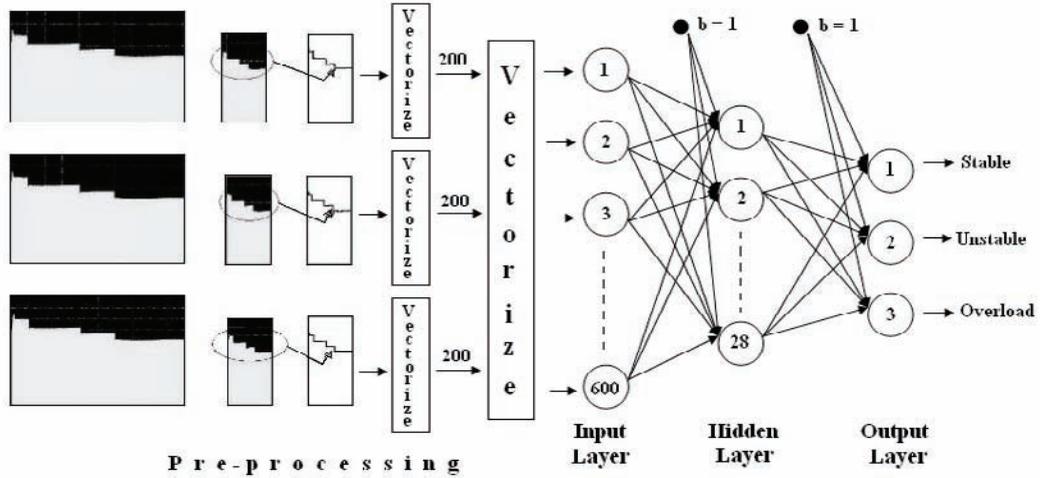
Table 1. Neural network final training parameters

Input Nodes	Hidden nodes	Output nodes	Learning coefficient	Momentum rate	Error	Iterations	Training time (seconds)	Run time (seconds)
600	28	3	0.001	0.33	0.002	12165	963*	0.02*

*using a 1.7 GHz PC with 256 MB of RAM, Windows XP OS and Matlab Programming Language

Table 2. Intelligent voltage instability detection results

Stable Case		Unstable Case		Overload Case		All Cases		
Training	Testing	Training	Testing	Training	Testing	Training	Testing	**Total**
10/10	7/8	10/10	8/8	10/10	8/8	30/30	23/24	**53/54**
(100%)	(87.5%)	(100%)	(100%)	(100%)	(100%)	(100%)	(95.83%)	**(98.1%)**

sures meaningful training while keeping the time cost to a minimum, and an output layer with 3 neurons representing the voltage output classification of stable, unstable or overload. During the learning phase, the learning coefficient and the momentum rate were adjusted during various experiments in order to achieve the required minimum error value of 0.002 which was considered as sufficient for this application. Figure 3 shows the topology of this neural network and the image pre-processing phase.

Implementation Results

The neural network learnt and converged after 12165 iterations and within 16 minutes (963 seconds), whereas the running time for the generalized neural network after training and using one forward pass was 0.02 seconds. Table 1 lists the final parameters of the successfully trained neural network. Voltage instability detection results using the training image set (90 images representing 30 cases) yielded 100% recognition as would be expected. The intelligent system implementation using the testing image set (72 images representing 24 cases that were not previously exposed to the neural network) yielded correct voltage output classification of 23 cases, thus achieving a 95.83% correct detection rate. Combining testing and training image sets, an overall recognition rate of 98.1% has been achieved. Table 2 shows the intelligent voltage instability detection results in details.

FUTURE TRENDS

The proposed system provides earlier detection of voltage instability. Upon the detection of the instability by the intelligent system, further measures can be taken to quickly sustain stability or clean voltage drop of the power distribution system. Future work will include the development of an intelligent voltage stabilizer that reads the classification output of our proposed intelligent detection system, and performs the necessary measures needed to stabilize the voltage output in case if unstable or overload case detection.

CONCLUSION

A fast and efficient intelligent system for detecting voltage instability in power distribution systems has been developed and presented within this article. Our hypothesis suggested that voltage output images of an assumed power system could be used to train a supervised neural network to classify the status of the power system voltage output.

The neural network within the intelligent system learnt within 963.4 seconds, whereas, the running time for the generalized neural network using one forward pass was 0.02 seconds. The reduction of training and generalization time was achieved by reducing the number of input patterns through processing the voltage output images, and adopting a unique method of extracting the input patterns using pixel positions. Here,

V

row numbers, at which grey level discontinuities occur, are found for each column in the voltage output graph and recorded for use as input patterns for the neural network implementation.

Our intelligent voltage instability detection system recognized correctly all training patterns as would be expected. Successful results were also obtained when using the testing patterns that were not exposed to the neural network before, yielding 95.83% correct detection. Table 2 showed in details the detection results, where the only incorrect classification of testing patterns occurred with a stable case that was classified as overload case. However, this single incorrect detection is not considered critically dangerous as it would be if, say, an unstable case was mistakenly classified as stable.

Finally, this article has proposed a different approach to detecting voltage instability in PDS by simulating a human operator's monitoring of voltage output graphs. Experimental results suggest that our method performs well and provides a fast and efficient system for voltage instability detection.

REFERENCES

Alcántara, F.J., & Salmerón, P. (2005). A New Technique for Unbalance Current and Voltage Estimation With Neural Networks. *IEEE Trans. Power Systems*. 20(2) 852-858.

Bansilal, Thukaram, D., & Harish Kashyap, K.(2003). Artificial Neural Network Application to Power System Voltage Stability Improvement. *IEEE International Conference on Convergent Technologies for the Asia-Pacific Region*. (1) 53-57.

Cardoso, G., Rolim, J.G., & Zürn, H.H. (2004). Application of Neural-Network Modules to Electric Power System Fault Section Estimation. *IEEE Trans. Power Delivery*. 19(3) 1034-1041.

De Souza, A.C.Z., Canizers, C.A., & Quintana, V.H. (1997). New Techniques to Speed up Voltage Collapse Computations using Tangent Vectors. *IEEE Trans. Power Systems*. 12(3) 1380-1387.

Ernst, D., Glavic, M., & Wehenkel, L. (2004). Power Systems Stability Control: Reinforcement Learning Framework. *IEEE Trans. Power Systems*. 19(1) 427-435.

Kamalasadan, S., Srivastava, A.K., & Thukaram, D.(2006). Novel Algorithm for Online Voltage Stability Assessment Based on Feed Forward Neural Network. *IEEE Power Engineering Society General Meeting*. Montreal Canada. 1-7.

Keyhani, A., Wenzhe, L., & Heydt, G.T. (2005). Neural Network Based Composite Load Models for Power System Stability Analysis. *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. 32-37.

Kundur, P., Paserba, J., Ajjarapu, V., Andersson, G., Bose, A., Canizares, C., Hatziargyriou, N., Hill, D., Stankovic, A., Taylor, C., Van Cutsem, T., & Vittal, V. (2004). Definition and Classification of Power System Stability. *IEEE Trans. Power Systems*. 19(2) 1387-1401.

Lin, H.C. (2007). Intelligent Neural Network-Based Fast Power System Harmonic Detection. *IEEE Trans. Industrial Electronics*. 54(1) 43-52.

Lukomski, R., & Wilkosz, K. (2003). Power System Topology Verification Using Artificial Neural Network Utilization of Measurement Data. *IEEE Power Tech Conference*. 180-186.

Michalewicz, K., Schmidt, M., Michalewicz, M., & Chiriac, C. (2005). Case Study: An Intelligent Decision-Support System. *IEEE Intelligent Systems*. 20(4) 44-49.

Mishra, S. (2006). Neural-Network-Based Adaptive UPFC for Improving Transient Stability Performance of Power System. *IEEE Trans. Neural Networks*. 17(2) 461-470.

Novosel, D., Begovic, M.M., & Madani, V. (2004). Shedding Light on Blackouts. *IEEE Power and Energy Magazine*. 2(1) 32-43.

Sjostrom, M., Cherkaoui, R., & Dutoit, B. (1999). Enhancement of Power System Transient Stability Using Superconducting Fault Current Limiters. *IEEE Trans. Applied Superconductivity*. 9(2)(1) 1328-1330.

Wenxin, L., Venayagamoorthy, G.K., & Wunsch, D.C. (2003). Adaptive Neural Network Based Power System Stabilizer Design. *IEEE Proc. International Joint Conference on Neural Networks*. (4) 2970-2975.

KEY TERMS

Artificial Neural Networks (ANN): A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in various applications such as robotics, speech recognition, signal processing, medical diagnosis, or power systems.

Back Propagation Algorithm: Learning algorithm of ANNs, based on minimizing the error obtained from the comparison between the outputs that the network gives after the application of a set of network inputs and the outputs it should give (the desired outputs).

Blackout: A power outage (complete collapse), a large-scale disruption in electric power supply.

Iterations: The number of epochs or repetitions of presenting a neural network with training input/output data.

Learning Coefficient: A numerical value that defines the learning capability of a neural network during training.

Momentum Rate: A numerical value that defines the learning speed of a neural network during training.

Pixel: A pixel (short for picture element, using the common abbreviation "pix" for "picture") is a single point in a graphic image.

Power Distribution System (PDS): Systems that comprise those parts of an electric power system between the sub-transmission system and the consumers' service switches. It includes distribution substations; primary distribution feeders; distribution transformers; secondary circuits, including the services to the consumer; and appropriate protective and control devices.

SCADA: A system that performs Supervisory Control and Data Acquisition, independent of its size or geographical distribution.

Voltage Instability: Voltage instability analysis is concerned with the inability of assessing the power system to maintain acceptable voltages at all system buses under normal conditions and after being subjected to disturbances (Kundur P. et al., 2004). A major factor contributing to voltage instability is the voltage drop that occurs when active and reactive power flow through inductive reactance of the transmission network. Voltage instability can be caused when a disturbance increases the reactive power demand beyond the sustainable capacity of the available reactive power resources.

Wave Reflection at Submerged Breakwaters

Alberte Castro Ponte
University of Santiago de Compostela, Spain

Gregorio Iglesias
University of Santiago de Compostela, Spain

Francisco Taveira Pinto
University of Porto, Portugal

Rodrigo Carballo
University of Santiago de Compostela, Spain

INTRODUCTION

Several types of structures are used in Coastal Engineering with the aim of preventing shoreline erosion, such as groynes, detached breakwaters, submerged breakwaters, etc. Submerged breakwaters have the advantage of their minimal visual impact, which has made them ever more popular (Chang & Liou, 2007).

When the incoming waves impinge on a submerged breakwater, a process of energy transformation occurs. Many laboratory and numerical studies have been carried out in order to investigate this process (Kobayashi & Wurjanto, 1989) (Losada, Losada & Martin, 1995) (Losada, Silva & Losada, 1996) (Liu, Lin, Hsu, Chang, Losada, Vidal & Sakakiyama, 2000). The energy of the incident wave is transformed as follows: (i) one part of this energy is transmitted above the crest of the structure and — in the case of permeable submerged breakwaters — through its interior; (ii) another part is dissipated by wave breaking and by friction with the structure during the transmission process and finally, (iii) the remaining energy is reflected seaward.

The reflection level is related with the scour in front of the structure. Therefore, a good knowledge about the reflection process may be helpful in order to avoid or at least mitigate the possible problems in the structure foundations. However, due to the complexity of the problem, the influence of all the relevant parameters (the structure slope and submergence, the water depth, the wave period and height, etc.) is not entirely understood yet and new approaches are needed.

In this work, an Artificial Neural Network (ANN) has been applied to a series of results obtained from a previous study of Taveira-Pinto (2001), in which several physical models were tested. Once trained and validated, the ANN has been used to estimate the wave reflection coefficient.

BACKGROUND

ANNs have proved to be a very powerful and versatile Artificial Intelligence technique (Orchad, 1993) (Haykin, 1999). In fact, they have been successfully applied to a great number of areas, including system identification and control, pattern recognition, data processing, time series prediction, modelling, etc (Rabuñal, Dorado, Pazos, Pereira & Rivero, 2004) (Rabuñal & Dorado, 2005).

In Civil Engineering, ANNs have been used most notably in Hydrology (Govindaraju & Rao, 2000) (Maier & Dandy, 2000) (Dawson & Wilby, 2001) (Cigizoglu, 2004). With regard to Ocean Engineering, ANN's have been applied to breakwater stability (Mase, Sakamoto & Sakai, 1995) (Medina, Garrido, Gómez-Martín & Vidal, 2003) (Kim & Park, 2005) (Yagci, Mercan, Cigizoglu & Kabdasli, 2005), wave forecasting (Tsai, Lin, & Shen, 2002) and tide-forecasting (Lee & Jeng, 2002).

ESTIMATION OF THE REFLECTION COEFFICIENT AT SUBMERGED BREAKWATERS

ANN Model

An Artificial Neural Network (Lippmann, 1987) (Haykin, 1999) is an information-processing system

consisting of an interconnected group of many simple process elements. These elements, also called neural units or neurons, work together in a similar way as biological neurons in the brain. The input is presented to the input neurons and propagated through the whole network until eventually some kind of output is produced.

In this work, a FeedForward Backpropagation network (FFBP) has been used. FFBP networks are composed of different layers of neurons linked by means of feedforward connections and trained by a back-propagation algorithm. Feedforward means that the output of a given neuron is used as the input of to the following layer, so there are no feedback loops. In this case, a network with two neuron layers, a logarithmic sigmoid hidden layer and a linear output layer, has been adopted.

The adjustment of the network weights in order to reduce the error is carried out by means of the back-propagation algorithm (Freeman and Skapura, 1991; Johansson et al., 1992). The error, *i.e.*, the difference between the network output and the target (the expected output) is propagated through the network backwards, up to the input layer; all the while the weights are tweaked. This process is repeated over and over until either the error is lower than a threshold or a maximum number of iterations are reached.

The ANN was trained by means of the Bayesian Regularisation method (MacKay, 1992), known to be effective in avoiding overfitting.

Experimental Set Up

The data used for training and testing the ANN were obtained in laboratory tests of submerged breakwaters

(Taveira-Pinto, 2001), carried out in the unidirectional wave tank of the Hydraulics Laboratory of the Faculty of Engineering of the University of Porto. The wave tank is 24.5 m long and 4.8 m wide with a maximum water depth of 0.40 m at the test section. The wave generator is a piston-type paddle, capable of generating regular and irregular waves. At the opposite end, a wave-absorbing gravel "beach" with a slope ratio of 1:20 was used in order to minimize the wave reflection level in the tank.

Six different impermeable models, constructed with wooden panels, were tested with different geometries (Fig. 1) at a 1:100 scale. Model height (h) was equal to 0.20 m in all cases (20 m in prototype). Different slopes (1:n) from 1:1 to 1:5 and two different crest widths (B), 0.05 m and 0.10 m, were tested.

Water surface displacements were measured using twin wire conductivity wave probes placed at different points in the wave tank. In order to evaluate the reflection coefficient (R), three wave probes were located on a line parallel to the wave direction. The spectral analysis method (Gilbert & Thompson, 1978), based on the Kajima (1969) method, was used to separate the incident and the reflection components.

A total of 275 tests were conducted with different water depths and irregular wave conditions. Water depths (d) between 0.20 m to 0.215 m, leading to freeboards (R_c) in the range 0 m to –0.015 m (negative for the breakwater crest below the still water level) were used during the tests. Irregular waves were generated conforming to the JONSWAP spectrum. Significant wave heights (H_s) from 2 cm to 8 cm and peak wave periods (T_p) from 0.8 s to 1.25 s were tested.

In order to carry out the training and the testing process of the ANN, the data were randomly divided

Figure 1. General layout of the testing models
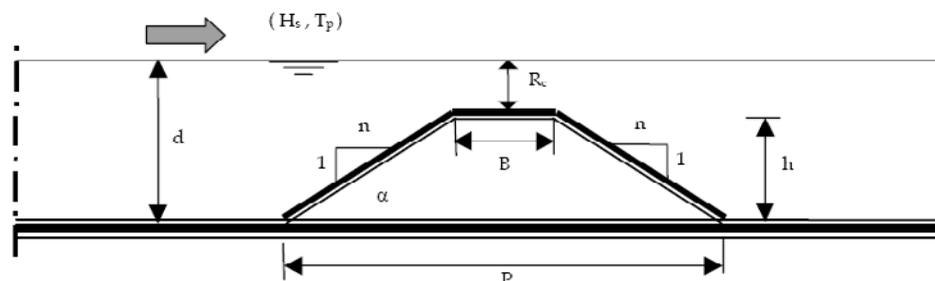
into a training data set (184 tests, or 67%) and a testing data set (91 tests, or 33%)

Both the geometrical parameters of the model and the wave spectrum parameters were introduced as inputs to the ANN by means of the following dimensionless numbers:

i. $\frac{R_c}{H_s}$ (relative freeboard)
ii. k_pB (relative crest width)
iii. k_pd (relative water depth)
iv. n (slope)

where k_p is the peak wavenumber obtained from the following expression:

$$\frac{4\pi^2}{T_p^2} = gk_p \tanh(k_pd)$$

The output of the ANN is the reflection coefficient R defined as the ratio between the reflected significant wave height and the incident significant wave height.

Training and Testing Process

The MSE obtained in the training and the testing of the ANN was 4.3×10^{-5} and 5.2×10^{-4} respectively. The small value of the testing error proves the ANN ability to generalize the knowledge acquired from the training data.

After the training process, the equation of the best linear fit to the data (Fig. 2) was $y = 0.9967x + 0.0011$, very close to the would-be perfect $y = x$. The value of the correlation coefficient $R^2 = 0.9973$ was also very good.

As was to be expected, the results of the testing process are slightly worse than those corresponding to the training process (Fig. 3). Nevertheless, both the coefficients of the regression equation ($y = 1.0364x - 0.0284$) and the correlation coefficient ($R^2 = 0.9853$) are very good.

ANN Application

Once trained and validated, the ANN was applied to analyze the influence of each input in the reflection process. In Fig. 4, the relative water depth (k_pd) is on the abscissa, and the reflection coefficient on the ordinate. Each curve corresponds to a constant freeboard value. The other two inputs were kept constant, the relative crest width at 0.2, and the slope at 1:2.

It means that the more the wave period the less the reflection coefficient in accordance with the experience which states that the long waves are related to higher reflection coefficients than the short waves.

The reflection coefficient decreases as the relative water depth (k_pd) increases. As for the relative freeboard, a smaller value (meaning more water over the breakwater crest) leads to a smaller reflection coefficient. In effect, the transmission process becomes

Figure 2. Regression analysis with the training data
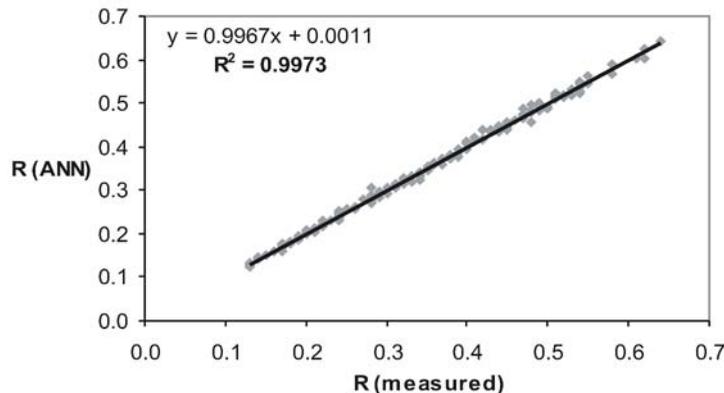
Figure 3. Regression analysis with the testing data
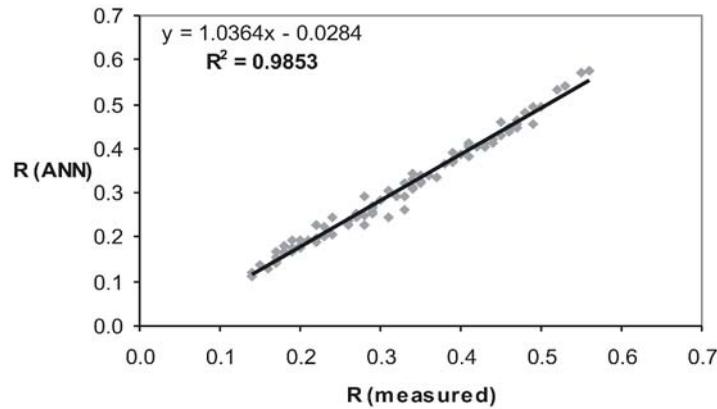
Figure 4. Variation of the reflection coefficient with the relative freeboard $(\frac{R_c}{H_s})$
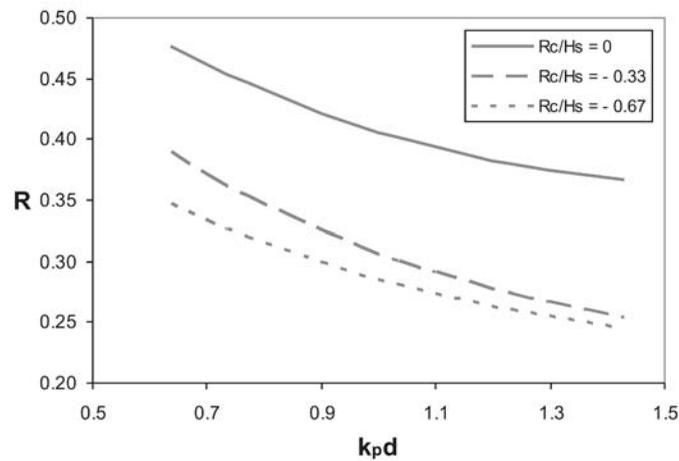
more efficient with higher submergence levels, leaving less energy for reflection.

The second parameter analyzed was the breakwater slope (Fig. 5). The influence of this input on the reflection coefficient can be easily explained. The limits of the slope value can be linked with a vertical wall ($n \to 0$) and a horizontal beach ($n \to \infty$). In the first case, the reflection is perfect and the reflection coefficient would be equal to 1. In the second, the reflection coefficient goes down to zero as the beach slope tends towards the horizontal. These trends are clearly showed on the graph.

Figure 5. Variation of the reflection coefficient with the breakwater slope (1:n)
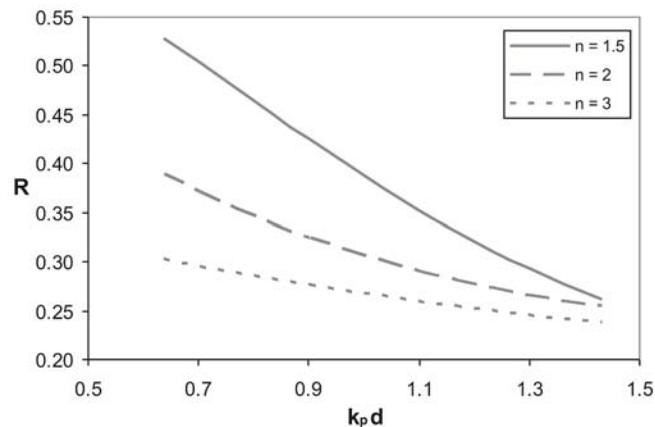
FUTURE TRENDS

ANNs have still a long way to go in Ocean Engineering applications. Both the stochastic nature of the wave action and the complexity of the energy transformation processes occurring when waves impinge on a structure lead to very intricate problems, which lend itself very well to ANNs. Hence it is to be hoped that the number of applications increases more and more in the coming years.

CONCLUSION

The estimation of the reflection coefficient at a submerged breakwater under the action of irregular waves is a very difficult task due to the great number of parameters involved: geometry and nature of the structure, water depth, significant wave height, wave period, etc. In this work, the behaviour of submerged breakwaters under the action of irregular waves was analyzed by means of a Feed-Forward Backpropagation network (FFBP), trained and tested on the basis of laboratory tests. The ANN model was shown to fit very closely the results of the physical model tests. The reflection coefficient diminished as the relative water depth increased. The effect of the model geometry is as follows. A decrease in the relative freeboard, meaning a

higher water level over the structure crest, brings about less reflection. As for the seaward slope, the reflection coefficient decreases with it. The curves drawn with resort to the ANN model not only help interpret these trends, but are an useful tool for the design engineer.

REFERENCES

Chang, H-K., Liou, J-C., 2006. Long wave reflection from submerged trapezoidal breakwaters. Ocean Engineering 34, 185-191

Cigizoglu, H.K., 2004. Estimation and forecasting of daily suspended sediment data by multilayer perceptrons. Advances in Water Resources 27,185-195.

Dawson, C.W., Wilby, R.L., 2001. Hydrological modeling using artificial neural networks. Progress in Physical Geography 25 (1), 80-108. 20

Freeman, J. A., Skapura, D. M., 1991. Neural Networks. Algorithms, Applications, and Programming Techniques. Addison-Wesley.

Gilbert, G., Thompson, D.M., 1978. Reflections in Random Waves: The Frequency Response Function Method, HR Wallingford, Report IT 173, Wallingford, UK

Govindaraju, R.S., Rao, A.R., 2000. Artificial neural networks in hydrology. Kluwer Academic Publishers, Dordrecth Boston, MA, p. 329.

Haykin, S. (1999). Neural Networks (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Johansson, E. M., Dowla, F. U., Goodman, D. M., 1992. Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method. Int. J. of Neural Systems, 2(4), 291-301.

Kajima, R., 1969. Estimation of incident wave spectrum in the sea area influenced by reflection, Japan, Society of Civil Engineers, Kyoto, Japon, Vol. 12, 9-16.

Kim, D.H., Park, W.S., 2005. Neural network for design and reliability analysis of rubble mound breakwaters. Ocean Engineering 32 (11-12), 1332-1349. 21

Kobayashi, N., Wurjanto, A., 1989. Wave transmission over submerged breakwaters. Journal of Waterways, Harbors, Coastal Engineering, ASCE 115, 662–680.

Lee, T.L., Jeng, D.S., 2002. Application of artificial neural networks in tideforecasting. Ocean Engineering 29 (9), 1003–1022.

Lippmann, R. P., 1987. An Introduction to Computing with Neural Nets, IEEE, ASSP Magazine.

Liu, P.L.-F., Lin, P., Hsu, T., Chang, K., Losada, I.J., Vida, L.C., Sakakiyama, T., 2000. A Reynolds averaged Navier–Stokes equation model for nonlinear water wave and structure interactions. In: Proceedings of the Coastal Structures '99, pp. 169–174.

Losada, I.J., Losada, M.A., Martin, F.L., 1995. Experimental study of wave-induced flow in a porous structure. Coastal Engineering 26, 77–98.

Losada, I.J., Silva, R., Losada, M.A., 1996. 3-D non-breaking regular wave interaction with submerged breakwaters. Coastal Engineering 28, 229–248.

Orchad, G., 1993. Neural Computing. Research and Applications. Ed. Institute of Physics Publishing, Londres.

MacKay, D. J. C., Bayesian interpolation, Neural Computation, vol. 4, no. 3, pp. 415-447,

Maier, H.R., Dandy, G.C., 2000. Neural network for the prediction and forecasting of water resources variables:

a review of modeling issues and applications. Environmental Modeling and Software 15, 101–124.

Mase, H., Sakamoto, M., Sakai, T., 1995. Neural network for stability analysis of rubble mound breakwaters. Journal of Waterway, Port, Coastal and Ocean Engineering, ASCE 121 (6), 294–299.

Medina, J. R., Garrido, J., Gómez-Martín, M.E., Vidal, C., 2003. Armour damage analysis using Neural Networks. Proc. Coastal Structures '03, Portland, Oregon (USA).

Rabuñal J.R., Dorado J., Pazos A., Pereira J., Rivero D., A New Approach to the Extraction of ANN Rules and to Their Generalization Capacity Through GP, Neural Computation, Vol. 16, pp. 1483-1524. 2004.

Rabuñal, J.R., Dorado J. (Eds.) Artificial Neural Networks in Real-Life Applications. Idea Group Inc. 2005.

Tsai, C.P., Lin, C., Shen, J.N., 2002. Neural network for wave forecasting among multi-stations. Ocean Engineering 29 (13), 1683–1695.

Van Oosten, R.P. and Marco, J, Peixó 2005. Wave transmission at various types of low-crested structures using neural netwoks, MSc Thesis, TUDelft.

Yagci, O., Mercan, D.E., Cigizoglu, H.K., Kabdasli, M.S., 2005. Artificial intelligence methods in breakwater damage ratio estimation. Ocean Engineering 32 (17-18), 2088-2106.

KEY TERMS

Artificial Neural Networks: Interconnected set of many simple processing units, commonly called neurons, that use a mathematical model representing an input/output relation.

Back-Propagation Algorithm: Supervised learning technique used by ANNs that iteratively modifies the weights of the connections of the network so the error given by the network after the comparison of the outputs with the desired one decreases.

JONSWAP Spectrum: Wave spectrum typical of growing deep water waves developed from field experi-

ments and measurements of waves and wave spectra in the Joint North Sea Wave Project.

Peak Period: The wave period determined by the inverse of the frequency at which the wave energy spectrum reaches its maximum.

Reflection: The process by which the energy of the incoming waves is returned seaward.

Significant Wave Height: In wave record analysis, the average height of the highest one-third of a selected number of waves.

Submerged Breakwater: Coastal protection structure crowned at, or below, the still water level.

Web-Based Assessment System Applying Many-Valued Logic

Sylvia Encheva
Haugesund University College, Norway

Sharil Tumin
University of Bergen, Norway

INTRODUCTION

The issue of rewarding partially correct answers has been addressed by many authors (Guzman, E. & Conejo, R., 2004, Gardner-Medwin, A.R. 1995, Huffman, D, Goldberg, F., & Michlin, M. 2003). Intelligent systems have been designed to assign scores related to the importance of missing or incorrect part of an answer. Such systems are meant to facilitate the process of knowledge assessment. While trying to be efficient in evaluating students' responses these systems operate with the answers to a single question addressing learning a new term, understanding a new concept or mastering a new skill. However, experimental practice shows that asking several questions about the same item results in inconsistent and/or incomplete feedback, i.e. some of the answers are correct while others are partially correct or even incorrect.

A large number of computer based systems and thus automated assessment systems lack the ability to reason with inconsistent information. Such a situation occurs when, f. ex. a student answers to two questions about one item and one of the answers is correct and the other one is incorrect or missing. Reasoning by applying classical logic cannot solve the problem because the presence of contradiction leads to trivialization, i. e. anything follows from 'correct and incorrect' and thus all inconsistencies are treated as equally bad (Priest, 2001).

In this paper we discuss how to assess students' understanding of new terms and concepts, shortly after they have been introduced in a subject. Application of **many-valued logic** allows the system to give meaningful responses in the presence of inconsistencies. Decision making rules, an intelligent agent is applying for assessing students' understanding of new terms and concepts are presented. Such rules distinguish between students' hesitation in the process of giving

an answer and lack of knowledge. We propose use of the generalized **Lukasiewicz**'s logic in a Web-based assessment system as a way of resolving problems with inconsistent and/or incomplete input.

BACKGROUND

A brief overview of a six-valued logic, which is a generalized **Kleene**'s logic (Kleene, S., 1952), has been first presented by Moussavi, M. & Garcia, N., 1989. Fitting, 1991 developed further this logic by assigning probability estimates to formulas instead of non-classical truth values.

The six-valued logic distinguishes two types of unknown knowledge values - permanently or eternally unknown value and a value representing current lack of knowledge about a state (Garcia, O.N. & Moussavi, M., 1990).

Two kinds of negation, weak and strong negation are discussed in Wagner, G., 1994. Weak negation or negation-as-failure refers to cases when it cannot be proved that a sentence is true. Strong negation or constructable falsity is used when the falsity of a sentence is directly established.

The semantic characterization of a four-valued logic for expressing practical deductive processes is presented by Belnap N.J., 1977. In Gurfinkel, A. & Chechik, M. 2005, it is shown that additional reasoning power can be obtained without sacrificing performance, by building a prototype software model-checker using **Belnap**'s logic.

Bi-dimensional systems representing and reasoning with temporal and uncertainty information have appeared also in Felix, P., Fraga, S., Marin, R., & Barro, S., 1999, and Mulsliner, D.J., Durfee, E.H., Shin, K.G., 1993.

A level-based instruction model is proposed by Park, C., & Kim, M., 2003. A model for student knowledge diagnosis through adaptive testing was developed by Guzman, E. & Conejo, R., 2004. An approach for integrating intelligent agents, user models, and automatic content categorization in a virtual environment is presented by Santos, C.T., & Osorio, F.S., 2004.

The Questionmark system at the University of Leeds applies multiple response questions where a set of options are presented following a question stem and the student can select any number and combination of those options. They are significantly more complex than multiple choice questions where the student can select only one among the suggested options. If a student marks some of the correct options (but not all) and or some of incorrect options his/her response can be correct, incorrect, partly correct or partly incorrect. The final outcome is correct or incorrect because the system is based on Boolean logic (Goodstein, R. L., 2007).

MAIN FOCUS OF THE CHAPTER

The test consists of two questions. According to the result of a test, understanding of a term or concept is achieved if a student gives a correct answer to questions about that term or concept. Such tests are placed after a new term or concept has been introduced in the theoretical part of a tutoring system. Questions in such tests should provide information about

- the student's knowledge,
- the subtler qualities of discrimination, judgement, and reasoning necessary in scientific reasoning,
- evaluate the student's judgement as to whether cause and effect relationships exist, and student's comprehension of a described situation.

Understanding of a Term

For evaluating understanding of a single term we propose a test where the choices can result in a correct answer, incorrect answer or unanswered question.

- Two correct answers imply understanding of that particular term. The process of questioning is terminated.
- One correct answer and one unanswered question imply some doubt about the student's understand-

ing of that particular term. The system first provides additional explanations and then suggests to the student to answer one new question taken from the database.
- One correct answer and one incorrect answer imply doubt about the student's understanding of that particular term. The system first provides additional explanations and then suggests to the student to answer two questions where one new question is taken from the database and the other question is taken from the first trial and has received an incorrect answer.
- Two unanswered questions imply uncertainty about the student's understanding of that particular term. The system first provides additional explanations and then suggests two new questions taken from the database.
- One incorrect answer and one unanswered question imply doubt about the student's understanding of that particular term. The system first provides additional explanations and then suggests to the student to answer the same questions.
- Two incorrect answers imply lack of understanding of that particular term. The system first provides additional explanations and then suggests to the student to answer the same questions plus one new question taken from the database.

If the second set of responses contains an incorrect answer and/or unanswered questions the system advises the student to work more with the originally provided learning materials and terminates the automated questioning process. We believe that several rounds of questioning would make the learning process time consuming for the student and thus disturb the learning flow.

However, the student can start a new assessment of his/her understanding of that particular term at any time he/she wants.

Understanding of a Concept

For evaluating understanding of a concept we propose a test with two questions where the choices can result in correct answer, partially correct answer, wrong answer or unanswered question.

- Two correct answers imply understanding of the concept. The process of questioning is terminated.
- One correct answer and one partially correct answer imply doubt about the student's understanding of the concept. The system first provides additional explanations and then suggests to the student to answer to the same question that has received a partially correct answer.
- One correct answer and one unanswered question imply doubt about the student's understanding of the concept. The system first provides additional explanations and then suggests to the student to answer one new question taken from the database.
- One correct answer and one wrong answer imply some doubt about the student's understanding of the concept. The system first provides additional explanations and examples, and then suggests to the student to answer again to the question that has previously received a wrong answer and one new question taken from the database.
- Two partially correct answers imply doubt about the student's understanding of the concept. The system first provides additional explanations selected theory and examples, and then suggests to the student to answer to the same questions.
- One partially correct answer and one unanswered question imply doubt about the student's understanding of the concept. The system first provides additional explanations and then suggests to the student to answer two new questions taken from the database.
- One partially correct answer and one wrong answer imply doubt about the student's understanding of the concept. The system first provides additional explanations, selected theory and examples, and then suggests to the student to answer two new questions taken from the database.
- One wrong answer and one unanswered question imply doubt about the student's understanding of the concept. The system first provides additional explanations selected theory and examples, and then suggests to the student to answer to the question that has previously received a wrong answer and a new question taken from the database.

Tests with a Larger Number of Questions

A test with three questions where the possible responses are correct answer, incorrect answer and a partially correct answer would require fifteen-valued logic. The generalized **Lukasiewicz**'s logic provides a solution for a test with any number of questions and answer options.

System Architecture

The system implementation is using the so-called **LAMP** Web server infrastructure and deployment paradigm. It is a combination of free software tools of an Apache Web server, a database server and a scripting programming platform on a Linux operating environment.

Behind this traditional three-tiers Web deployment is a service support sub-system. Communication framework based on **XML-RPC** is used to connect the Web application middle-ware and the intelligent assessment/diagnostic system together. The separation of these two units made it possible to modularly design and implement the system as loosely couple independent sub-systems.

The dynamic page publisher compiles a page to be presented to the user from a template file in relation to the user response, current state variables and activities history. A template file contains the static declarations of a document. The variables in a particular template files are given values by the dynamic page publisher module during the production of an HTML document. The resulting HTML document is sent back to the user Web browser. This module also acts as a handler when a user requests a page or sends a form back to the Web server.

The users stack profiler keeps track of user activities history in a stack like data structure in the database. Each event, like for example response/result of a test or a change of learning flow after following a hint given by the system, is stored in the database. This module provides the percept to the intelligent modules of the software agents' sub-system. The users stack profiler communicates directly with the agents by sending messages over the **XML-RPC** communication channel. By using some common data stored in the database, the users stack profiler indirectly affects the behaviour of the user's agents and visa verse.

The application middleware and the software agents run independently of each other. As such, they can be situated on different servers. The middleware implement the Web side of the system while the software agents implement the decision side of users learning process. Given a certain response to a particular test at a particular user state, what best action can be taken to increase the probability that the user will learn a particular unit of knowledge? This decision is done by the intelligent diagnostics agent.

The intelligent assessment agent does an early diagnostic about absorption of knowledge. A response given by a particular student from a test will give the system an indication about the state of learning of a particular term. This agent helps to implement a part of an intelligent tutoring system, which differ from the intelligent diagnostics agent. The intelligent assessment agent facilitates students' early absorption/assimilation of new terms.

FUTURE TRENDS

The proposed assessment system is based on many-valued logic. Further research is needed to investigate which of the available non-classical logics can provide more accurate assessments according to what is the subject of that assessment - knowledge of a term, understanding of a concept, level of mastered skill, etc.

Another important area for future work involves recommendation, of hints, explanations, examples, and theory, tailored to each student's responses and needs.

CONCLUSION

This paper is devoted to assessing students understanding of new terms and concepts. The presented framework provides flexibility in the choice of logic and can serve as an effective exploration tool for reasoning about many combinations of input coming from different sources.

REFERENCES

Belnap, N.J. (1977). A useful four-valued logic, In *Modern uses of multiple-valued logic*, J.M. Dunn and G. Epstain (eds), D. Reidel Publishing Co., Dordrecht, 8-37.

Felix, P., Fraga, S., Marin, R., & Barro, S., (1999). Linguistic representation of fuzzy temporal profiles, *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 7(3), 243-257.

Fitting, M. (1991). Kleene's Logic, Generalized. *Journal of Logic and Computation.* 1(6), 797-810.

Garcia, O.N. & Moussavi, M., (1990). A Six-Valued Logic for Representing Incomplete Knowledge, Proc. of the 20th International Symposium on Multiple-Valued Logic (ISMVL), *IEEE Computer Society Press*, Charlotte, NC, USA, 110-114.

Gardner-Medwin, A.R. (1995). Confidence assessment in the teaching of basic science. *Association for Learning Technology Journal.* 3, 80-85.

Goodstein, R. L., (2007). Boolean Algebra. *Dover Publications*

Gurfinkel, A. & Chechik, M. (2005). Yasm: Model-Checking Software with Belnap Logic. *Technical Report* 470, University of Toronto.

Guzman, E. & Conejo, R. (2004). A model for student knowledge diagnosis through adaptive testing. *Lecture Notes in Computer Science*, 3220, 12-21.

Huffman, D, Goldberg, F., & Michlin, M., (2003). Using computers to create constructivist environments: impact on pedagogy and achievement. Journal of Computers in mathematics and science teaching, 22(2), 151-168.

http://www.leeds.ac.uk/perception/v4_mrq.html

Kleene, S., (1952). *Introduction to Metamathematics*. D. Van Nostrand Co., Inc., New York, N. Y.

Moussavi, M. & Garcia, N., (1989). A Six-Valued Logic and its application to artificial intelligence, Proc. of the Fift Southeastern Logic Symposium, *IEEE Computer Society Press*, UNC-Charlotte, NC, USA.

Moussavi, M. & Garcia, N., (1989). A Six-Valued Logic and its application to artificial intelligence, Proc. of the

Fift Southeastern Logic Symposium, *IEEE Computer Society Press*, UNC-Charlotte, NC, USA.

Mulsliner, D.J., Durfee, E.H., Shin, K.G., (1993). CIRCA: A coop erative intelligent real-time control architecture. Trans. on Sytems, Man and Cybernetics, 23(6), 1561-1574.

Park, C., & Kim, M. (2003). Development of a Level-Based Instruction Model in Web-Based Education. *Lecture Notes in Artificial Intelligence,* 3190, 215-221.

Priest, G., (2001). *An Introduction to Non-Classical Logic*, Cambridge Press.

Santos, C.T., & Osorio, F.S., (2004). Integrating intelligent agents, user models, and automatic contentcategorization in virtual environment. *Lecture Notes in Computer Science,* 3220, 128-139.

Wagner, G., (1994). Vivid Logic: Knowledge Based reasoning with two kinds of negation, *Lecture Notes in Artificial Intelligence*, 764.

KEY TERMS

Belnap's Logic: It has four truth values 'T, F, Both, None'. The meaning of these values can be described as follows: an atomic sentence is stated to be true only (T), an atomic sentence is stated to be false only (F), an atomic sentence is stated to be both true and false, for instance, by different sources, or in different points of time (Both), and an atomic sentences status is unknown. That is, neither true, nor false (None).

Kleene's Logic: Kleene's logic has three truth values, truth, unknown and false, where unknown indicates a state of partial vagueness. These truth values represent the states of a world that does not change.

LAMP Web Server: It is a combination of free software tools of an Apache Web server, a database server and a scripting programming platform on a Linux operating environment.

Lukasiewicz's Three-Valued Logic: Lukasiewicz's three-valued valued logic has a third value, 1/2, attached to propositions referring to future contingencies. The third truth value can be construed as 'intermediate' or 'neutral' or 'indeterminate'.

Lukasiewicz's Generalized Logic: It is done by inserting evenly spaced division points in the interval between 0 and 1.

Six-Valued Logic: The six-valued logic obtained as an extension of the Kleene's logic has six truth values - *true*, *false*, *unknown*, $unknown_t$ - intermediate level of truth between *unknown* and *true*, $unknown_f$ - intermediate level of truth between *unknown* and *false*, *contradiction.*

XML-RPC: It is remote procedure calling using HTTP as the transport and XML as the encoding.

Workflow Management Based on Mobile Agent Technology

Marina Flores-Badillo
CINVESTAV Unidad Guadalajara, Mexico

Ernesto López-Mellado
CINVESTAV Unidad Guadalajara, Mexico

INTRODUCTION

Nowadays Information Systems (IS) are designed for individual task execution control allowing coordinating, monitoring, and supporting the logistical aspects of a *business process*, in other words, the IS has to manage the flow of work through the organization.

The WorkFlow Management represents a critical issue for achieving enterprise competitiveness among organizations. Many companies have realized that the *business processes* (*BP*) within their organizations, and between the companies and their partners have not been clearly described and there are not enough techniques and methods to automate the processes.

The *Workflow Management Coalition* (*WFMC*) states that *workflow* (WF) is concerned with the automation of procedures where documents, information, or tasks are passed to the participants according to a defined set of rules to achieve, or contribute to, an overall business goal (WfMC, 1999). Another definition of WF can be found in (Rusinkiewicz & Seth, 1994) where *workflows* are *activities* involving the coordinated execution of multiple tasks performed by different processing entities (persons or machines). A *task* or process involves a piece of work and a process entity which executes the work.

Workflow Management (WFM) is a fast evolving technology which is increasingly being exploited by businesses in a variety of industries. Its primary characteristic is the automation of processes involving combinations of human and machine-based activities (Aalst & Hee, 2002), (Aalst, 1998).

A *Workflow Management System* (WFMS) provides procedural automation of a *business process* by management of the sequence of work activities and the invocation of appropriate human and/or IT resources associated with the various activity steps. Although the most prevalent use of *WFMS* is within the office environment in staff intensive operations such as insurance, banking, legal and general administrations, etc, it is also applicable to some classes of industrial and manufacturing applications (WfMC, 1995). *WFMS* needs to integrate other technologies such that *agent* technology, which provides flexible, distributed, and intelligent solutions for business process management.

This work presents a methodology for mobile agent-based WFMS development. The proposed methodology consists of a modular and gradual specification of the system where a mobile agent guides the process through organizational units and executes different tasks. Several mobile agents evolve through the system executing concurrently their assigned task.

BACKGROUND

Workflow Management

The notion of *agent* in (Yuhong, Zakaria & Weiming, 2001) is used as "a computer system situated in some environment, which is capable of autonomous action in this environment in order to meet its design objectives" (different notions can be found in (Wooldridge, 2002) and (Nwana, 1996)). These works also highlight the benefits of applying agent technology to *business process* management; some of these benefits are: distributed system architecture, the inherent autonomy of software agents because agents can start a WF based on event trigger, the agent reactivity because it have the ability to generate alternative execution paths, etc. An intelligent agent is capable of autonomous operation and flexible behavior in order to meet its design goals and also has the properties of reactivity, pro-activity, and social ability (Wooldridge, 2001).

In other works both concepts are integrated. In (Repetto, Paolucci & Boccalette, 2003), a methodol-

ogy for the design of agent based WF was presented; it consisted in three steps. In the first step the authors model the *BP* with UML Activity diagrams by identifying all the necessary resources and activities. In the second step, all the activities identifying roles in parallel paths are grouped. Finally, they define an agent for each group.

Several researchers took the *agent* technology for the improvement of WF applications. In (Marin & Brena, 2005), an architecture for high-level agent-based WF is proposed. On this architecture they break down the WF execution and the process flow control in small execution units handled by intelligent agents and a WF processes is controlled in a decentralized way.

A collaborative approach for workflow systems is presented in (Savarimuthu & Purvis, 2004) where agents collaborate by forming social network (societies), in (Savarimuthu, Purvis & Fleurke, 2004) agents are embedded in a system that can monitor and control the overall functioning of a workflow process in an agent based WF system.

In (Minhong, Huaiqing & Dongming, 2005), agent technology is used for the WF monitoring where various intelligent agents working together to perform flexible monitoring tasks in an autonomous and collaborative way.

Multi-Agent Systems

Mobile agents are autonomous programs that can travel from one computer to another under its own control. They offer a robust and efficient framework to develop distributed applications including mobile applications.

A stationary agent is executed only on the system where it began its execution. If it requires information from a different system or needs to interact with another agent, it uses a standard client-server communication (RMI, RPC, CORBA).

A mobile agent (MA) is not always attached within the systems where it starts the execution, rather it is capable of moving itself through the network nodes where it is allowed, modifying eventually its execution environment; the MA carries with itself its current state and its code (strong mobility). Furthermore, MAs may exhibit several advantageous features due to mobility, for example a) interaction with the resource during its migration to the needed resource location, keeping the bandwidth and reducing the latency of the network

(Cabri, Leonardi & Zambonelli, 1998), b) interaction with the users during the migration to the user location, answering faster user requests. In both cases the agent continues the interaction with the resource or the user even with temporary network connections failures.

Most of distributed applications fit naturally on the model of MAs because the agents can migrate sequentially through a computer network, they send other agents to visit computers in a parallel way, they remain stationary and interact with remote resources, etc.

There exist several organizations with the aim of establishing standards for agent software development and agent interoperability. One of them is FIPA (FIPA, 1997).

JADE Development Tool

JADE (Java Agent Development Framework) is a software framework completely implemented in Java language which simplifies MA system implementations by using a middleware which fulfill FIPA (FIPA, 1997) specifications. The agent platform can be distributed through machines (which not necessary share the same OS) and the configuration can be managed by a remote GUI (Bellifemine, Caire, Trucco & Rimassa, 2006).

The communication architecture offers flexible and efficient message passing where JADE creates and manage the incoming private ACL message queue for each agent. The complete FIPA communication model has been implemented and its components have been clearly distinguished. JADE integrates completely interaction protocols, ACL, ontology's, transport protocols, etc. Most of the FIPA defined protocols are available in JADE.

MOBILE AGENT-BASED WFMS DEFINITION

This work presents a methodology for the development of Mobile Agent-based *WFMS*. The basic idea for conceiving such a system is that a MA guides the workflow process through the different organizational units in which several *tasks* are executed according to the handled *case*.

During the design phase the components are described in a clear and compact way. The system is described as a set of interconnected organizational units that have a specific resource allocation. The agent

Figure 1. Structuring the design of a WfMS
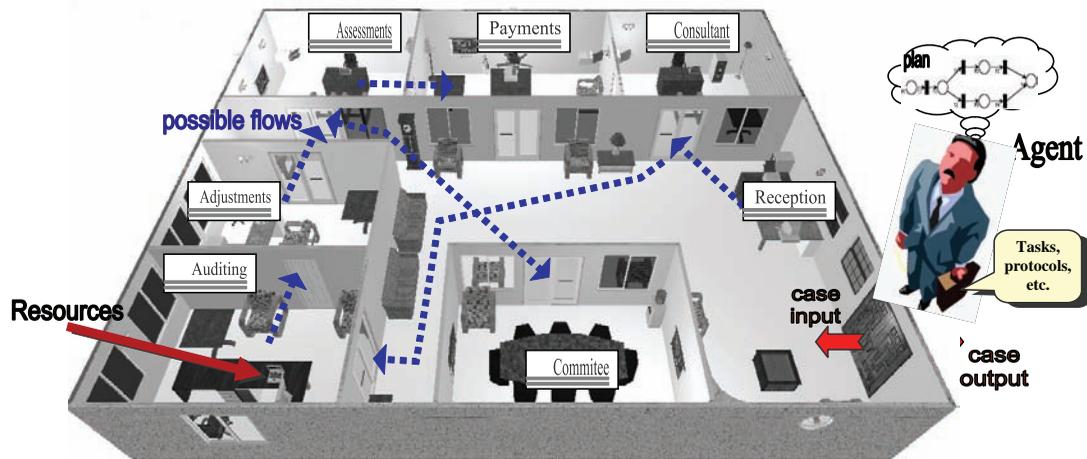
behavior is determined by two kinds of specifications: a) the description of the agent general behavior and common knowledge for all the agents, namely, basic operations, and interaction protocols (collaboration, and resource competition); and b) particular descriptions of a specific behavior such as the *task* plan and an accessibility roadmap, which describe the assigned process and the permitted access to the organizational units respectively. This strategy is illustrated in figure 1.

The implementation phase is supported by a software development guideline allowing the definition of Java components (using also the middleware JADE) from agent systems specification from the design phase. The obtained software is distributed in a set of networked computers that manages MA migration. The modularity allows adaptations to system specification changes without difficulties.

For the sake of readability the proposed method is illustrated through a case study dealing with claim processes in an insurance company.

Case Study Description

The problem can be defined as follows: "*Define the WF for the claim processes in an insurance company in which a customer claims the insurance policy of a personal property (real state, car, life insurance). The company must receive the claim, request personal data from the customer (insurance policy number, etc.), and validate the insurance validity, payments and beneficiaries. It must do the adjustment of real damages, validate the case, calculate the correspondent assess-*

ment, do the necessary payments to the customer if the complain is valid, or inform in case that the process has some invalid data "

Design Methodology

The MA that guides the process through the company must have the previous knowledge of organizational units allocation, resource allocation, the execution plan, the list of *task*s to perform and, the different needed protocols for the resource solicitation or competition. Additionally the MA must know the environment structure, which is first specified.

Environment specification. The agent environment is defined by the diagram that represents the general structure of the company. It is necessary to identify the different departments in which some task is executed or the information flows, considering all the possible *cases*. Then each department or office is represented by a JADE agent container where we can suppose that each container is on a different host of a distributed network system. In this way each department is represented by a host. A host can belong to different platforms, but for simplicity work we suppose that all hosts belong to the same platform.

Here we propose a strategy for the platform definition but this can have a different distribution. So, for the platform distribution it is important to:

- Identify the departments involved with the process considering all the possible *cases*.

Figure 2. Claim insurance process a) Block diagram of the organizational units b) Environment Definition
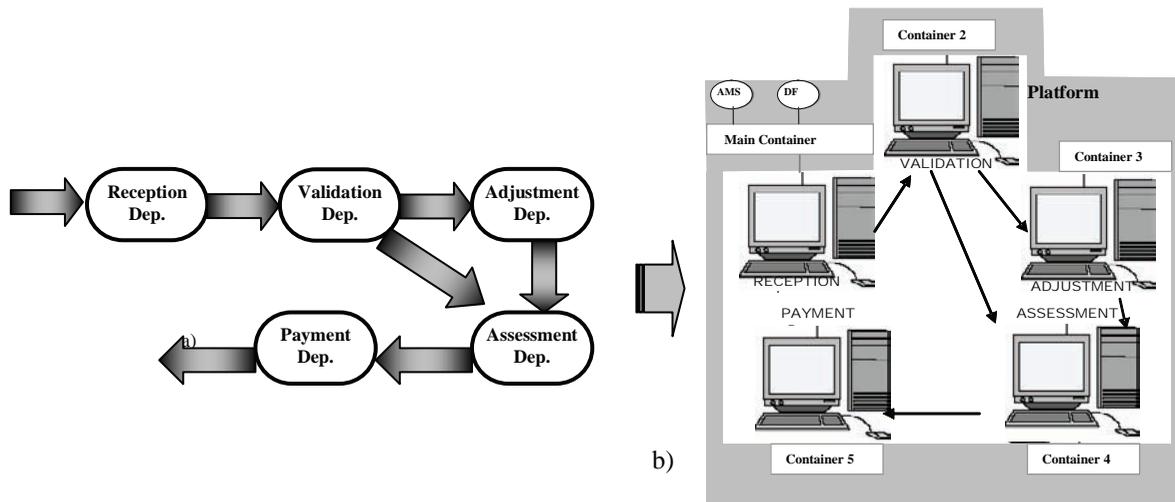
- Identify the information flow and its direction.
- Construct a block diagram with the obtained information.

In this diagram, each block represents a site or host of a distributed system (each host has a JADE agent container) and the arrows represent the direction of the flow or a possible agent migration.

Consider that in our case of study there are five departments: reception, validation, assessment, adjustment and payment; we can get the blocks diagram and the possible platform configurations as the figure 2.

In JADE the containers creation is achieved as follows:

- Main Container, on commands line of the site that are going to have this particular kind of container we can write:

C:\ java JADE.Boot   -container-name Name_Host [-gui]

where each [] represents an optional parameter.
- The rest of the containers are created using the line:

C:\ java JADE.Boot -container-name Name_Host -container –host HostMainContainer

- We can also use the GUI of JADE for the creation of containers and agents

Mobile Agent Definition. For the case study, the states of the agent general behavior can be easily represented for the Petri net showed in fig. 3. The MA selects a plan execution according to the *WF process definition* for the assigned case; the plan indicates which sites the agent must visit in order to process the case, an access map for the sites, information for resource reservation. Also the agent migrates from one site to another, collaborates with other agents, competes for resource allocation, etc.

In JADE a mobile agent is created as a sub-class of the generic class *Agent* and its service is registered with the DF of JADE. We can use the code in Exhibit A.

The particular behavior for this agent is given for the WF process definition. In other words, is defined for the order execution of the tasks involved in the process of the case. For the programming of the plan we can use the available Behaviours of JADE because they represent the tasks that an agent can perform. We can use any of the different behaviours included in JADE according to the plan to perform.

The plan is obtained from the *WF process definition*; it defines the execution sequence of the involved tasks; so, sometimes it is necessary to construct a diagram (like a flow diagram) specifying sequence when it includes

Exhibit A.

```
public class MobileAgent extends Agent {
protected void setup() {
DFAgentDescription dfd = new DFAgentDescription();
dfd.setName(getAID());
ServiceDescription sd = new ServiceDescription();
sd.setType(Type);
sd.setName(Name);
dfd.addServices(sd);
try {
DFService.register(this, dfd);
}catch (FIPAException fe) { fe.printStackTrace(); }
  } }
```

Figure 3. Petri Net for the Agent general behavior
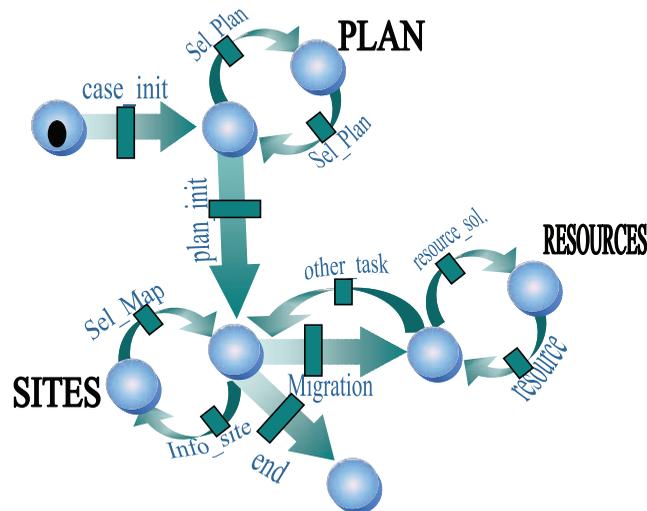
alternatives; the diagram must indicate in which department the task must be performed or executed.

Following the case study, assume that the obtained diagram is that shown in figure 4; in this figure both the task name and the corresponding department are indicated.

If the execution sequence of tasks follows the behavior of a Finite State Machine (FSM) we use the JADE FSMBehaviour.

```
FSMBehaviour fsm = new FSMBehaviour(this) {
public int onEnd() {
System.out.println("FSM behaviour completed");
myAgent.doDelete();
return super.onEnd();}  };
```

If we use this particular behavior we have to register the appropriated states which represent each task to perform, and the transitions that represent its sequence or its execution order. In this way, for the register of states it is used the function:

registerState(Function_Name_Taski, state_namei),

where the first parameter indicates the name of a function for the correspondent task and the second parameter indicates a name for this state used for the transitions registration.

For this we use the function:

RegisterDefaultTransition(state_name1, state_name2),

which indicates that after the complete execution of the function represented for the state *state_name1* the function for the state *state_name2* is performed.

Some states and transitions for the case study could be those showed in figure 4. Also one must add other states (*MovValBehaviour, MovValoBehaviour, MovCotizBehaviour* y *MovPayBehaviour*) for the migration of the agent when the performed task must be realized in a department different to the current.

Because of the agent migration the appropriated mobility ontology must be registered:

```
getContentManager().registerLanguage(new SLCodec(), FIPAN-ames.ContentLanguage.FIPA_SL0);
// register the mobility ontology
getContentManager().registerOntology(MobilityOntology.get-Instance());
```

When the states for the FSMBehaviour are registered we use only a name for the function to perform; so, it is necessary to add the Java statements for each function for the selected behavior. Each one of these methods is added as a class that inherits of one of the JADE *Behaviours*. These methods contain the statements to execute in each task; for the case study a method definition is included in figure 4.

If the agent has to collaborate with another agent to perform a task, the JADE protocol FIPA-Request can be used (an example is shown in figure 5).

Following the guidelines given above the general behavior of the MA and the environment where it evolves can be defined. In a similar way the behavior of stationary agents can be also established.

The proposed methodology has been applied to several case studies leading to modular software, which has been executed on several networked (LAN) personal computers. The tests were performed on sites in which a JADE platform was defined. Nevertheless different configuration platforms can be integrated in the WFMS.

FUTURE TRENDS

The proposed methodology for the development of WFMS is a first step towards the automation of complex business processes in large enterprises. However in companies where the organizational units are distributed

in several cities, the MA must travel though the web; then more sophisticated capabilities must be added to the agents and their environment, namely security protocols and agent losses control.

Furthermore, it would be advantageous consider the interaction with existing WFMS based on the standards provided by WFMC to profit of existing information and business strategies.

CONCLUSION

Automation of business processes yields improvements to the productivity of companies. This work proposed a methodology for developing WFMS based on mobile agent technology. This methodology allows modular definitions for the environment and mobile agent behaviors. The proposed implementation technique uses JADE getting all the JAVA advantages. The mobile agent can interact with other agents to collaborate, negotiate or compete for resources. Due to the modularity of the obtained software, it can be easily modified according to modifications to WFMS specifications.

REFERENCES

Aalst, W.M.P.v.d., and Hee, K.v. (2002) Workflow Management: Models, Methods and Systems. London, MIT Press.

Aalst, W. van der (1998). The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers.

Bastin Tony Roy Savarimuthu, Maryam Purvis, and Martin Fleurke. Monitoring and controlling of a multi-agent based workflow system. In Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalization, pages 127-- 132. Australian Computer Society, Inc., 2004.

Bellifemine F., Caire G., Trucco T., Rimassa G. (2006). **JADE** Programmer's **Guide**. available at: http://JADE.tilab.com/doc/programmersguide.pdf

Cabri G., Leonardi L., Zambonelli F. (1998), Mobile Agent Technology: Current Trends and Perpectives, Congresso annuale AICA'98, Napoli (I), November 1998.

Figure 4. Identification of states and transitions for the FSMBehaviour


```
private class RegSolBehaviour extends OneShotBehaviour{
 MessageTemplate mt;
public void action(){
        mt= MessageTemplate.MatchPerformative(ACLMessage.INFORM);
        ACLMessage msg = myAgent.blockingReceive(mt);
        try{
                Package1 p1= (Package1) msg.getContentObject();
                UserAgentAID=p1.AgentAID;
        sComplaint=p1.ComplaintNm;
                }catch(UnreadableException e3){
                    System.err.println("catched exception "+e3.getMessage());
        }
        //send message to user and begin the proccess of the complaint
        ACLMessage reply =  new ACLMessage(ACLMessage.REQUEST);
        reply.addReceiver(UserAgentAID);
        reply.setInReplyTo(msg.getReplyWith());
        reply.setContent(getAMS().toString());
        reply.setConversationId(sComplaint);
        reply.setReplyWith(sComplaint+System.currentTimeMillis());
        myAgent.send(reply);
}//end Behaviour
```

Reception Dep.
Complain register

fsm.registerFirstState
(new RegSolBehaviour(),St_RegSol);

Validation Dep.
Data Validation

Valid ?

No

YES

Current dep.
Data register

fsm.registerLastState
(new RegBehaviour(),St_Reg);

fsm.registerState
(new ValDamaBehaviour(),St_ValDama);

Adjustment dep.
Adjustment

fsm.registerDefaultTransition(St_Pay,St_Reg);

Proceed?

No

fsm.registerTransition (St_ValDama,St_Reg,NOVALID);

fsm.registerTransition
(St_ValDama,St_MovCotiz,VALID);

Y

fsm.registerState
(new CotizBehaviour(),St_Cotiz);

Assessment Dep.
Assessment claim

Payment Dep.
payments

fsm.registerState
(new PayBehaviour(),St_Pay);

Figure 5. Fragment of the FIPA-Request protocol

FIPA (1997). Foundation for Intelligent Physical Agents, Specifications. Available at: http://www.fipa.org.

Hollingsworth David, (1995). Workflow reference model. Technical report. The Workflow Management Coalition, Document Number WFMC-TC-1003. available at: http://www.wfmc.org/

Marin Cesar A., Brena Ramon (2005), Multiagent Architecture for Decentralized Workflow Process Execution, Technical Report, Center for Intelligent Systems, Tecnológico de Monterrey.

Minhong Wang, Huaiqing Wang, Dongming Xu (2005), The design of intelligent workflow monitoring with agent technology, Knowledge-Based Systems, Volume 18, Issue 6, pages 257-266.

Nwana, Hyacinth (1996). Software Agents: an Overview, Knowledge Engineering Review, Vol. 11, No 3, pp 1-40.

Repetto Marco, Paolucci Massimo y Boccalatte Antonio (2003). A design tool to Develop Agen-Based Workflow Management Systems, Proc. Italian Workshop, from Objets to Agents: Intelligent Systems and Persuasive Computing (WOA2003), Villasimius, Italy

Rusinkiewicz M., Sheth A. (1995). Specification and execution of transactional workflows. In W. Kim, editor, Modern Database Systems: The Object Model, Interoperability and Beyond, pages 592--620. ACM Press, New York, NY.

Savarimuthu, B.T.R and Purvis, M. (2004). A Collaborative Multi-Agent Based Workflow System, Knowledge-Based Intelligent Information and Engineering Systems, M. Negoita, R. J. Howlett, and L. C. Jain, (eds.) ISSN: 0302-9743, Lecture Notes in Artificial Intelligence (LNAI), vol. 3214.

WfMC. (1999) Workflow Management Coalition - Terminology & Glossary. Technical report, The Workflow Management Coalition, Document Number WFMC-TC-1011, available at: http://www.wfmc.org/

Wooldridge, Michael (2001). Intelligent Agents: The Key Concepts, Multi-Agent-Systems and Applications 2001, p 3-43, Springer-Verlag, Berlin, Heidelberg

Wooldridge Michael (2002), An Introduction to multiagent Systems, John Wiley & Sons (Chichester, England) ISBN 0 47 149691X. 340 pp.

Yuhong Yan, Zakaria Maamar, Weiming Shen (2001), Integration of Workflow and Agent Technology for Business process Management, The sixth international Conference en CSCW in Design, London, Ontario, Canada.

W

KEY TERMS

Activity: A description of a piece of work that forms one logical step within a process. An activity may be a manual activity, which does not support computer automation, or a workflow (automated) activity.

Agent: An agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.

Business Process: A set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.

Case: The representation of a single enactment of a process, using its own process instance data, and which is (normally) capable of independent control and audit as it progresses towards completion or termination.

Mobile Agent: A program that can migrate from a computer to other computer within a heterogeneous network. The program chooses when and where to migrate. It can suspend its execution at an arbitrary point, transport to another computer and resume execution in the new computer.

Multi-Agent System: Is a collection of software agents that work in conjunction with each other. They may cooperate or they may compete, or some combination of cooperation and competition.

Process: A formalized view of a business process, represented as a co-coordinated (parallel and/or serial) set of process activities that are connected in order to achieve a common goal.

Process Definition: The representation of a business process in a form which supports automated manipulation, such as modeling, or enactment by a workflow management system.

Workflow: The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.

Workflow Management Coalition: The WFMC is a non profit organization with the objectives of advancing the opportunities for the exploitation of workflow technology through the development of common terminology and standards.

Workflow Management System: A system that completely defines creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.

Index

Symbols

2 Pages graph layout problem, definition 1120

2-D data representation, definition 1438

2D objects with GNG 1365

2-D representation, solar radiation data 1433

2D-PAGE analysis 1583–1588

2Pages graph layout problem 1117

3D deformable model, definition 630

3D object recognition (3DOR) 248

3D reconstruction tests 843

A

absorbance, definition 588

ACO 1531

acoustic attenuation spectrum, definition 1302

acoustic systems, optimization of 1296

action planning 1549

action planning, definition 1554

active appearance models (AAM), definition 630

active interactive genetic algorithms (aiGAs) 788

active learning 1, 1383

active learning, with SVM 1–8

active model, definition 553

actuators, definition 955

adaptation 592

adaptation, definition 470

adaptive business intelligence 16–21

adaptive computation paradigm 9

adaptive geometric computing 9

adaptive neural algorithms 22–30

adaptive neuro-fuzzy systems 31–36

adaptive rule-driven devices 37

adaptive spatial memory (ASM) 9

adaptive system, definition 580

adaptive technology 37–44

adaptivity 37

additive monotonic cooling 346

ad-hoc networks, evolutionary computing approach 589

ad-hoc networks, swarm intelligence approach for 1530

admissible alternative, definition 436

advanced cellular neural networks image processing 45–50

affectation step, definition 1251

AGALZ 1316

agent applicability 952

agent autonomy, definition 955

agent based mode, definition 1237

agent based query answering system 925

agent learning process 928

agent percepts, definition 955

agent taxonomy 941

agent typology 941

agent, definition 513, 931

agent-based modeling 51–57

agents, types of 951

agglomerative hierarchical clustering 231

AI 918

AIBO, definition 630

aIGA weight tuning 788

air flight control, definition 540

ambient intelligence (AmI) 85

ambient intelligence (AmI) environment 92–98

ambient intelligence (AmI), in elderly healthcare 94

ambient intelligence (AmI), in tourism 94

ambiguous grammar, definition 773

AmI, in vehicles and transports 94

analogic cellular computer 45

analytic modelling 969

angiographies 110–117

angiographies, and image segmentation 110–117

ANN 1358, 1360, 1361, 1568

ANN application 1605

ANN development with genetic programming 619

ANN development, evolving graphs 618

ANN model 1603

ANN simplification, evolving graphs 618

annotation, definition 342

ANNs design, evolutionary approaches for 575

ANNs modeling, mathematical framework for 1057

ANNs modeling, mathematical methods for 1059

ANNs output combination 556

ANNs, need of use 1358

ANOVA 304

answer set programming (ASP) 1373

ant colony algorithm 242

ant colony optimization, definition 1535

AntHocNet 1532

AntHocNet, definition 1535

anti-granulometries 1107

architecture, definition 1017

area of the search space, definition 493

arrhythmia, definition 916

ART (adaptive resonance theory), definition 1496, 1503

ART, definition 1242

articulation ontology, definition 499